La Commode



CH. BLANCHER

35 Francs

Le Magazine des Ordinateurs Commodore



58 rue Notre-Dame-de-Lorette - 75009 PARIS
Tél. : 282.19.80 - Télex : 290350 F

Boutique

Essayez !

Tous matériels annoncés disponibles à l'essai.







Réfléchissez !

Un véritable service avant et après-vente.

- Les conseils du spécialiste pour le matériel et le logiciel.
- Dépannages rapides par notre service après-vente intégré.
- Une garantie 1 an pièces et M.O. sur tous les produits.

Comparez !

Des micro-prix sur tous les micros.

Extrait de notre tarif. Prix TTC.

MATERIELS	
CBM 4032	9990,00 F
CBM 4040	9900,00 F
CBM 4022	5700,00 F
CBM 8032	13300,00 F
CBM 8050	13300,00 F
CBM 8026	14500,00 F
CBM 8027 SANS CLAVIER	11100,00 F
IMPRIMANTE CBM 8024	12250,00 F
IMPRIMANTE SEIKOSHA GP 80 + INT. CBM	3640,00 F
VIC 20	2350,00 F
NOUVEAUTÉS VIC	
RAM 3K	290,00 F
MODULATEUR N/B	190,00 F
LECTEUR K7	520,00 F
DISQUE 5'	4150,00 F
IMPRIMANTE	3050,00 F
MONODISQUE 4000/8000	4400,00 F

LOGICIELS	
TRIEX 8000	1050,00 F
MASTER 8000	2800,00 F
ASSEMBLEUR 8000	810,00 F
PASCAL 8000	1100,00 F
COMPTABILITE 8000	4000,00 F
PAIE 8000	2800,00 F
TRAITEX 8000	2800,00 F
VISICALC 8000	1100,00 F
OZZ 8000	3400,00 F
ASSEMBLEUR 4000	800,00 F
TRAITEX 4000	2800,00 F
FICHIER MAILING 4000	760,00 F
AGENTS ASSURANCE	3000,00 F
VISICALC 4000	1100,00 F

P.S.I.	
VISA POUR L'INFORMATIQUE	45,00 F
LA REALISATION DES PROGRAMMES	50,00 F
PROGRAMMER EN ASSEMBLEUR	75,00 F
PROGRAMMER EN BASIC	65,00 F
LE BASIC ET SES FICHIERS	75,00 F
PROGRAMMER EN LSE	65,00 F
PROGRAMMER EN PASCAL	75,00 F
COMMENT PROGRAMMER	75,00 F
LA DECOUVERTE DE L'APPLESOFT	65,00 F
LA PRATIQUE DE L'APPLE II VOL. 1	65,00 F
LA PRATIQUE DE L'APPLE II VOL. 2	65,00 F
LA PRATIQUE DE L'APPLE II VOL. 3	75,00 F
LA PRATIQUE DU MZ 80K	65,00 F
LA DECOUVERTE DE PET/CBM	65,00 F

LA PRATIQUE DU PET/CBM VOL. 1	65,00 F
LA PRATIQUE DU PET/CBM VOL. 2	75,00 F
LA PRATIQUE DU TRS 80 VOL. 1	65,00 F
LA PRATIQUE DU TRS 80 VOL. 2	85,00 F
LA PRATIQUE DU TRS 80 VOL. 3	75,00 F
THE ZX 80 POCKET BOOK	65,00 F
LE PETIT LIVRE DU ZX 81	65,00 F
LA DECOUVERTE DU PC 1211	75,00 F
LE LANGAGE ADA	75,00 F
PROGRAMMER EN APL	65,00 F

DOCUMENTATION GÉNÉRALE SUR DEMANDE.

En raison des fluctuations monétaires ces prix sont susceptibles d'être modifiés sans préavis. Nous consulter pour confirmation.

HARD communication

JCR, l'informatique service compris.

Sommaire

COMMODORE

un vrai micro-ordinateur

En choisissant le Commodore VIC 20 pour entrer dans l'univers fabuleux de la micro-informatique, vous acquérez un ordinateur évolutif qui est en mesure de devenir, au fur et à mesure de l'accroissement de vos connaissances ou de vos besoins, un véritable petit système informatique. Petit certes par la taille, mais aux possibilités remarquables qui lui permettent de vous offrir un rapport prix/performances actuellement sans équivalent.

Merveilleux outil d'initiation à l'informatique, compact, tout en étant très puissant, le Commodore VIC 20 se branche sur un moniteur TV ou sur votre téléviseur*.

Simple d'utilisation, le VIC 20 met la micro-informatique au service de l'enseignement, de la formation, des calculs scientifiques et techniques, des loisirs, etc.

***Moniteur ou sortie téléviseur**

23 lignes de 22 caractères.

- □ moniteur monochrome (vert et blanc ou noir et blanc)
- □ TV noir et blanc (avec adaptateur)
- □ TV SECAM utilisable en noir et blanc (avec adaptateur)
- □ TV couleurs PAL ou bi-standard PAL/SECAM
- □ TV couleurs SECAM (avec adaptateur couleur).

puissant et évolutif

Le VIC 20 est fréquemment équipé d'un qui permet de conserver des programmes et des données sur cassettes. Evolutif, le VIC 20 peut se muscler de différentes façons : modules de mémoire complémentaire, unité de monodisquette, imprimante, etc.

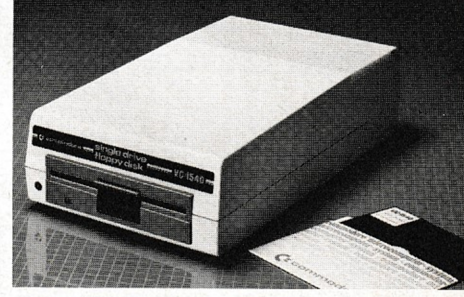
Unité de monodisquette VIC 1540

L'unité de monodisquette VIC 1540 permet d'exploiter au maximum toutes les possibilités du VIC 20. Elle offre une méthode de stockage de programmes et de données où se combinent vitesse, sécurité et facilité d'accès.

Grâce à l'utilisation conjointe de l'unité de disque et de l'imprimante, le VIC 20 devient un véritable système informatique.

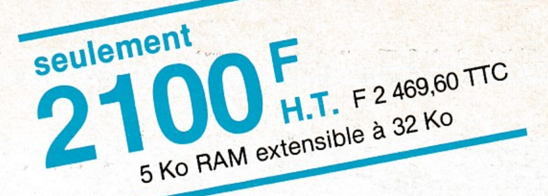
Principales caractéristiques :

- 170 K octets de capacité,
- disque 5 pouces 1/4, simple densité (sectorisation par logiciel),
- connexion directe, sans interface, avec le VIC 20,
- compatibilité directe avec l'imprimante VIC 1515,



- périphérique « intelligent » : sa connexion ne diminue en rien la mémoire disponible sur le VIC 20,
- compatibilité directe avec les unités de disques CBM 2031 et CBM 4040.

VIC 20



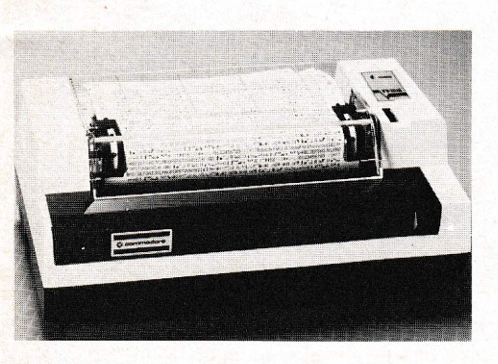
aux étonnantes possibilités

Imprimante graphique VIC 1515

L'imprimante VIC 1515 a été spécialement conçue pour le VIC 20.

Elle peut être utilisée pour effectuer des impressions de programmes, des lettres, éditer des tableaux, tracer des graphiques...

Principales caractéristiques :
- 80 caractères par ligne,
- impression à aiguilles,
- 30 caractères par seconde,
- tous les caractères du VIC,
- caractères élargis,
- caractère programmable,
- impression graphique.



BASIC

Le VIC 20 dispose d'un Basic étendu qui a fait la réputation des Commodore.

OFFRE SPECIALE

pour configuration autoformation au Basic : au lieu de 2 910 F H.T. . . . **2 720 F H.T.**
comprenant : 1 VIC 20 + 1 lecteur de cassettes + 1 programme autoformation au Basic

Initiez-vous à l'informatique
autoformation assistée par ordinateur



Assisté continuellement par l'ordinateur, vous apprenez à votre rythme.

Une méthode complète

Le cours d'autoformation au Basic sur micro-ordinateur VIC 20 est constitué d'un manuel et de deux cassettes contenant une série de programmes.

Le cours est divisé en quinze unités. Chaque unité comporte une leçon théorique et un travail pratique à réaliser sur votre micro-ordinateur. Pour commencer, vous branchez l'ordinateur sur un moniteur TV ou votre téléviseur. Ainsi, votre travail sera visualisé sur l'écran.

Puis, vous introduisez une des cassettes dans le magnétophone relié à l'ordinateur.

Une méthode active

A partir du texte de la leçon et des exemples de programmation mémorisés sur cassette, vous vous initiez progressivement au maniement du micro-ordinateur.

A la fin de chaque série, un questionnaire d'auto-évaluation des connaissances acquises vous permet de mesurer vos progrès. Mémorisé sur cassette, ce test s'affiche à l'écran du téléviseur. Vous tapez vos réponses au clavier du micro-ordinateur qui en vérifie l'exactitude. Vous savez donc toujours où vous en êtes dans vos études.

Un outil pédagogique efficace

Utilisant toute la puissance du micro-ordinateur évolutif VIC 20 de Commodore — qui offre un rapport prix/performances remarquable — tout a été mis en œuvre pour que le cours d'autoformation au Basic soit un outil pédagogique efficace.

L'outil qu'il vous faut pour vous

connaissance sera l'un de vos meilleurs atouts pour affronter l'avenir.



PROCEP MAXI SERVICES POUR LA MICRO

Nombreuses extensions complémentaires

Cartouches extensions mémoire.
Programmer Aid (VIC 1212).
Machine language Monitor (VIC 1213).
Super expander (VIC 1211 M).
Module d'extension (VIC 1010).
Interface RS232C (VIC 1011).
Interface IEEE-488.
Accessoires de commande de jeux.

DEMANDE DE DOCUMENTATION

à adresser à votre Distributeur-Revendeur Commodore/Procep ou à Procep, 19-21, rue Mathurin-Régnier, 75015 PARIS.

Oui, je désire en savoir plus sur les étonnantes possibilités du Commodore VIC 20. Envoyez-moi votre brochure en couleurs.



Nom ______________________ Prénom ______________________

N° __________ Rue ______________________________________

Code Postal ______________________ Ville ______________________

SFC C 04.82

commodore

EDITORIAL

Ça y est, nous sortons les numéros à la bonne date! Enfin direz-vous, mais La Commode a beaucoup changé sa présentation qui, nous l'espérons vous satisfait.

Un événement d'importance a marqué ces derniers mois, il s'agit du FORUM COMMODORE-PROCEP qui a permis de faire le point des matériels et des logiciels. La rubrique actualité lui est consacrée.

A cette occasion, vous avez été nombreux à venir nous voir pour échanger des idées sur la Commode et exposer vos préoccupations. Nous créons ce mois-ci, une nouvelle rubrique "LE FEED BACK DE LA COMMODE" où vous pourrez juger les articles sur le fond et sur la forme, et nous permettre d'améliorer VOTRE revue.

Dans ce numéro vous trouverez deux articles de base sur la programmation : MIEUX UTILISER BASIC et OPERATIONS SUR LES FICHIERS. Ces articles de fond sont destinés à vous aider, aussi n'hésitez pas à nous écrire pour nous envoyer vos problèmes et vos critiques.

En attendant de vous voir au SICOB 82, toute l'équipe de La COMMODE vous souhaite de BONNES VACANCES.

Jean-Luc DESCHAMPS

Actualité

FORUM ... FORUM ... FORUM ... FORUM

Le FORUM 82 qui s'est tenu du 13 au 15 Mai dernier à PARIS, est un événement important pour les "Commodoristes". D'une part, le fait de démarrer une exposition annuelle uniquement consacrée à la gamme Commodore, d'autre part, le nombre des exposants et la qualité des produits présentés donne une bonne idée de la vitalité de Commodore.

LES MATERIELS

La gamme complète de Commodore était présente :

le VIC 20 avec ses extensions et ses programmes,
le 4000 avec l'unité monodisque souple 2031, et
le 8000 très professionnel avec son extension à 96K,
l'unité de disques durs 20 Mega octets (P8140), la carte CP/M, le graphique haute résolution (512-256). Une nouveauté l'imprimante CBM 8023, légèrement plus lente que la 8024, mais moins encombrante et avec des possibilitées graphiques que n'avait pas sa grande soeur.

8140 : c'est une unité de disque dur de deux fois 10 Mega-octets avec un disque amovible, ce qui permet une sauvegarde de tout repos. Nécessite un CBM 8096 et le logiciel M/DOS 6502. Ce système n'est pas relié au CBM par le bus IEEE, mais par l'extension mémoire pour des raisons de rapidité.
(prix : 59500 Frs H.T. avec M/DOS 6502).

CBM 8023 : imprime à 150 c/s avec 136 car. par ligne en mode normal et 230 en mode compressé. Entraînement à traction et à friction.

Contrairement à la 8024, elle a le jeu de caractères graphiques, et en fait imprime comme la 4022 avec en plus, le mode binaire qui, lié à l'option GRAPHEX, permet la recopie d'écran (30 secondes pour un écran de 512 x 256).

TRANSNEX : deux contrôleurs de transmissions, SYNCHRONE OU ASYNCHRONE, en versions coffret pour CBM 8032/8096 avec 8050. A noter que TRANSNEX aynchrone existe en version carte à monter à l'intérieur du CBM.

(prix : TRANSNEX asynchrone
carte : 2950 Frs H.T.
coffret: 3450 Frs H.T.
TRANSNEX synchrone :
coffret : 14500 Frs H.T.

CP/MAKER : pour les fanatiques du CP/M, le CBM dispose maintenant de cette possibilité. C'est une carte contenant un Z80 et 64K de mémoire vive qui est rajoutée. Deux modes possibles :

- Mode Commodore Standard, qui ignore la carte CP/M

- Mode CP/M, c'est alors le Z80 qui travaille en utilisant le 6502 du CBM 8032 comme processeur d'entrée/sortie.

Ce dispositif livré avec une disquette CP/M 2.2 pour CBM 8032, sera disponible prochainement pour 3000 et 4000. (prix :5450 Frs H.T.)

Mais attention, comme chez tous les constructeurs, il n'y a pas d'intercheangeabilité des disquettes 5 pouces.

GRAPHEX : carte graphique haute-résolution permettant 512 x 256 points (prix : 3950 Frs H.T.).

Extension mémoire pour CBM 8032 : permet de le transformer en 8096 (prix : 2950 Frs H.T.)

LES LOGICIELS

De ce côté, un foisonnement impressionnant de logiciels de très haut niveau :

- MDOS/6502 (voir P8140)
- CP/M (déjà vu)
- PM 96 (partition mémoire pour 8096)
- MASTER 96 (version 96 K)
- SILICON OFFICE
- MANAGER

Ceci pour les logiciels présentés par PROCEP. Mais sur les autres stands, on pouvait trouver un grand nombre de programmes allant du compilateur BASIC aux programmes de gestion d'un cabinet d'assurance !

Mais, reprenons ces logiciels dans l'ordre.

MDOS/6502

Fait pour un CBM 8096 muni du disque dur de 2 fois 10 Mega octets, ce logiciel regroupe en fait trois parties :

- Gestion de fichiers.
  fichiers séquentiels, relatifs, à accès par clé, multiclés.

- Gestion d'écran.
  Par masque de saisie avec contrôle intégrés.

- Gestion des éditions
  Par masques d'éditions.

PM 96

Il s'agit d'un système de gestion de l'extension mémoire du 8096, écrit en langage machine pour permettre des programmes BASIC jusqu'à 86K (28K variables, 58K programme)

MASTER 96

Imaginez un programme de gestion d'écran de 2 pages en BASIC réduit à quelques lignes et vous aurez une idée de la puissance de MASTER.

C'est en fait l'outil indispensable à tous les utilisateurs de CBM qui développent des logiciels commercialisés. En effet, MASTER, indépendamment de ses possibilitées de programmation, comprend un dispositif de protection par clé électronique.

SILICON OFFICE

Tout en un ! ou le "bureau électronique".

Un programme qui réside entièrement en mémoire (8096 !), et dont le but est de résoudre tous les problèmes des PME/PMI. Pour cela, on dispose de 36 commandes de génération de programmes, 26 commandes de traitement de texte et 14 fonctions de calcul.

Cet ensemble permet :

- le stockage et la recherche d'informations.
- la création, l'édition et l'impression de textes.
- des calculs mathématiques.
- la transmission des informations.

MANAGER

Il s'agit d'un système de ges-

tion de fichiers très élaboré, fait pour faciliter toutes les tâches d'acquisition, de stockage et de manipulation des informations. A noter qu'il peut être utilisé conjointement avec TRAITEXT.

Sur les stands, on pouvait remarquer :

Chez A.I.O. (Assistance Informatique de l'Ouest, un programme de TRI en ROM pour CBM 8096 ; un programme de gestion de bibliothèque.

Chez Computer Shop JANAL : la possibilité de transformer un CBM 4016 en 4032 et de l'étendre à 80 colonnes (4016 grand écran s'entend).

Chez Télé Systèmes, France Câble radio, la présentation du système COMMODIAL qui fonctionne avec un TRANSNEX Asynchrone et permet un accès aux grandes banques de données (QUESTEL), au système d'échange et de stockage (EURODIAL) et au service de messagerie électronique (MISSIVE).

Jean-Luc DESCHAMPS

RECTIFICATION

Le prix des Monodisques

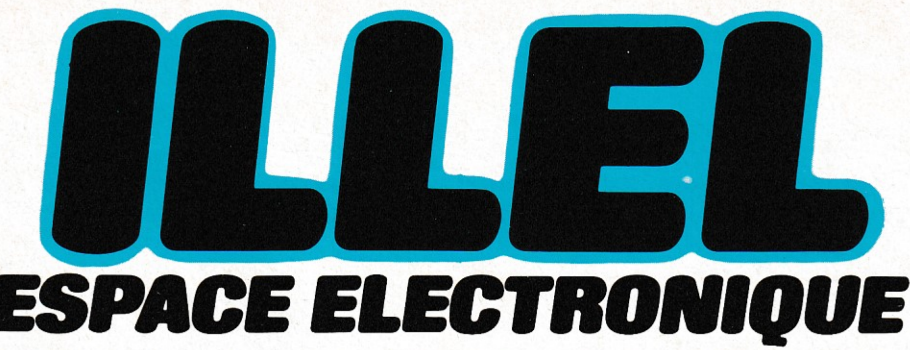
Compte tenu des éléments qui étaient en notre possession lors de la rédaction du numéro 3, nous avions prévu un prix trop élevé pour les monodisques 2031 et nous avions d'ailleurs indiqué que, à notre avis, cela faisait une différence trop importante par rapport à l'unité VIC 1540.

Nous ne pensons pas que c'est dû uniquement à notre protestation, mais quoi qu'il en soit, les prix sont maintenant :

VIC 1540 : 3750 Frs H.T.
4410,00 Frs TTC
(17,60, attention au changement de taux)

CBM 2031 : 3950 Frs H.T.
4645,20 Frs TTC

Il faut noter que 2x2031 < 4040 (8950 Frs H.T.). Nous parlerons du choix "deux 2031 ou un 4040 ?" dans un prochain numéro, car il y a des raisons de préférer l'une ou l'autre solution.



ESPACE ELECTRONIQUE

ILLEL CENTER PARIS 15e : 143, av. Félix-Faure - 75015 Paris, Tél. 554.97.48, Métro : Balard.
ILLEL CENTER PARIS 10e : 86 bd Magenta 75010 Paris, Tél. 201.94.68, Métro : Gare de l'Est, Parking : Magenta.

commodore

Système CBM 8001

PME/PMI, services décentralisés ou autonomes des grandes entreprises, professions libérales..., vous qui avez besoin de puissance pour des applications professionnelles de gestion, de bureautique, d'instrumentation, de process industriel, etc., voici votre système informatique : le Système CBM 8001. Toujours à un prix micro, mais doté de capacités qui en font un outil puissant apte à résoudre vos applications professionnelles.
Le Système CBM 8001 associé à des logiciels de haut niveau (logiciels "prêts à l'emploi" : comptabilité, paie, etc., ou des logiciels "ouverts" : Ozz, Visicalc, etc.) apporte une gestion efficace aux PME/PMI tout en étant un précieux auxiliaire pour les cadres des services décentralisés ou autonomes des grandes entreprises.

Micro-ordinateur CBM 8032

Écran 80 colonnes, 2 000 caractères. Écran incorporé à affichage très fin. Éditeur d'écran : "scroll" avant et arrière de l'image. Définition de fenêtres de travail sur



l'écran. Clavier type machine à écrire, qwerty ou azerty. Clavier numérique séparé. Microprocesseur 6502 de MOS Technology (Commodore). Mémoire RAM disponible 32 K octets. Basic étendu, résident ; gestion de fichiers. Moniteur langage machine résident. Accès au langage machine par le Basic Interface IEE-488. Port de 8 lignes d'entrées/sorties parallèles.

Unité de double minidisquette CBM 8050

14.053 F TTC

Capacité : 1 million d'octets. Une unité de 2 disquettes de 512 000 octets. Enregistrement simple face simple densité. Périphérique "intelligent" avec : 2 microprocesseurs, 16 buffers d'entrées/sorties (4 K octets de mémoire RAM). DOS résident sur 16 K octets de



ROM. La mémoire vive du micro-ordinateur est ainsi intégralement disponible.

Imprimante CBM 8024

L'imprimante CBM 8024 est connectée directement au bus IEEE-488 de l'unité centrale sans aucune



extension ou interface supplémentaire. Elle est gérée par microprocesseur. Mécanisme d'impression : TALLY MANNESMANN. Matrice 7 x 7 (option 9 x 9). Jeu de 96 caractères (majuscules et minuscules).

COMMODORE Série 4000

Micro-ordinateur CBM 4032

Écran 25 lignes, 40 colonnes. Écran incorporé à affichage très fin. Clavier 73 touches. Microprocesseur 6502 de MOS Tchnology (Commodore). Mémoire RAM 32 K octets. Basic étendu, résident. Gestion de fichiers. Moniteur langage machine résident. Accès au langage machine par le Basic. Port de 8 lignes d'E/S parallèles. **10.000 F TTC**

Micro-ordinateur CBM 4016 :

Caractéristiques identiques au CBM 4032. Mémoire RAM : 16 K octets.



Unité de double minidisquette CBM 4040

Capacité : 340 K octets. Une unité de 2 disquettes de 170 000 octets utiles par disquette. Enregistrement simple face simple densité. Périphériques "intelligents" avec 2 microprocesseurs, 16 buffers d'entrées/sorties (4 K octets de mémoire RAM). Système d'exploitation (DOS) intégré sur 16 K octets de ROM dans l'unité de disquettes. La mémoire vive du micro-ordinateur est ainsi intégralement disponible. **10.525 F TTC**

Imprimante CBM 4022

80 colonnes. Imprimante à aiguilles. Mécanisme d'impression EPSON. Entraînement du papier par tracteurs à picots. Impression des caractères alphanumériques et semi-graphiques du CBM. Matrice 5 x 8. 1 original + 2 copies.



COMMODORE VIC 20

Mémoire 3,5 K extensible 27,5 K. Interface cassette. Basic étendu commodore. Écran (23 lignes x 22 caractères). **2 490 F TTC**

LOGICIELS

PET SOFT . (Liste sur demande)

CBM 3001

TRAITEMENT DE TEXTE (Français) . . . . . . 1 117,20 F TTC
GESTION FICHIER CLIENT ET MAILING . . 764,40 F TTC
PROGRAMME DE PAIE . . . . . . 1 117,20 F TTC
PROGRAMME COMPTABILITÉ 1 117,20 F TTC
PROGRAMME CABINET D'ASSURANCES 3 057,60 F TTC
GESTION DES VENTES . . 1 940,40 F TTC

CBM 8001

OZZ :
Logiciel d'écriture de programmes d'applications, demandant seulement l'entrée des paramètres de travail . . . . 3 469 F TTC

VISICALC :
Un puissant outil de planification et de prévision . . . . . 1 117 F TTC

COMPTABILITÉ 8000 :
Logiciel de haut niveau structuré pour non seulement traiter votre comptabilité, mais aussi répondre aux attentes des experts-comptables.
A partir du traitement des journaux habituels, ce logiciel vous permet :
- la saisie en mode conversationnel des pièces comptables sur un bordereau dessiné à l'écran ;
- l'enchaînement programmé des travaux comptables entraîne la mise à jour automatique des comptes pour chaque pièce saisie . . . . 4 116 F TTC

TRAITEMENT DE TEXTE Pour automatiser les travaux dactylographiques, frappe, contrôle, mise en page, corrections, modifications . . . . 2 881 F TTC

diagonale

BON DE COMMANDE EXPRESS ILLEL

A retourner à : ILLEL Center Informatique : service vente par correspondance 143, avenue Félix Faure 75015 Paris.

MODE DE RÈGLEMENT CHOISI

☐ à la commande paiement comptant
☐ à crédit* à partir de 2000 F.
Je verse 20 % du montant total de mon achat : ______,___ F ci-joint : Chèque bancaire ☐ C.C.P. ☐ Mandat carte ☐
* Conditions de crédit CREG : • Être salarié, • 20 % minimum au comptant, solde arrondi à la centaine supérieure.

Je soussigné : Nom ______ Prénom ______
N° ______ Rue ______
Code postal ______ Ville ______ Tél. ______
commande ferme et désire recevoir d'urgence

	Quantité	Prix unitaire	Prix total

Signature :

Montant net
Frais de port pour envoi postal 30,00
TOTAL A PAYER

Courrier des Lecteurs

Je vous écris à la suite de l'article FCCF paru dans la Commode numéro 3.

Je suis très intéressé par cette fédération de club, et aussi je souhaiterais obtenir de plus amples renseignements.

J'envisage de créer un club de micro-informatique Commodore dans ma région, et dans ce but, je voudrais quelques conseils.

Pourriez-vous publier une liste des clubs qui vous auraient contactés pour faire partie de la FCCF?

Je vous remercie d'avance, et continuez dans votre effort de présentation, et d'amélioration.

Jacques BRUGEASSOU
73400 UGINE

*
* *

Je possède un CBM 3032, et je suis abonné à la "Commode" depuis son deuxième numéro. Je suis vraiment satisfait en ce qui concerne cette revue, bien que le délai de parution me paraisse bien long. J'ai lu votre troisième numéro avec délice ! Je voudrais cependant apporter une petite modification en ce qui concerne l'article sur le bricolage d'un bouton RESET ; en effet, je pense que la réalisation de cette modification bien utile ne me paraisse dangeureuse si l'on ne possède pas de bases en électronique. Une mauvaise soudure pourrait devenir fatale pour notre charmant petit compagnon... C'est pourquoi, je vous propose cette solution :

1) Installation de Reset :

Regardons de plus près les connecteurs J-9 et J-4 situés sur la droite du PET/CBM ; pour ce bouton, on branche un fil sur la broche A-22, et la masse sur la broche B-22 (Notons d'ailleurs que B-1 A B-25 sont à la masse du connecteur J-4). On relie ainsi ces deux fils sur un bouton poussoir.

2) Installation de NMI

On effectue ce branchement sur la broche A-24 et B-24 du même connecteur J-4.

3) Installation du Diag Sense :

Même système, on branche un fil sur la broche A-20 et la masse sur la broche B-20, mais attention! Il faut effectuer ce branchement sur le connecteur J-9 cette fois!!!

Pour une réalisation plus propre de ces branchements, il est préférable d'acheter deux connecteurs, ou alors, souder les fils sur des petites pines simples qui se branchent sur chaque broche...

On a ainsi la même réalisa-

tion que dans le numéro 3, sans avoir à bidouiller des choses qui peuvent être fatales à notre animal domestique préféré...

J'espère que cette remarque vous intéressera ; elle évite en tout cas de faire des bêtises!!!

J'ai, avec cette remarque, plusieurs questions à vous poser, et qui me tiennent à coeur depuis assez longtemps :

- Je voudrais savoir quand exactement l'unité 2031 sera en vente sur le marché français ;

- Vous avez parlé dans le numéro 3 de la création d'une bourse d'échange de programmes ; est-ce-qu'il serait possible d'échanger des programmes et des logiciels dont nous ne sommes pas l'auteur (comme "RABBIT", EXTRAMON, etc.)

- Comment peut-on créer un club Commodore, membre de la "FCCF" à Nice ?

- Y aura t-il un autre salon Commodore l'année prochaine, car j'ai reçu le numéro 3 qui en parlait, 3 jours après la fermeture de ce forum !!!, et j'aurai bien aimé aller le visiter...

- Pourrait-on créer une rubrique petites annonces, concernant la recherche et la vente de matériels Commodore, ou de produits adaptables au Commodore ?

- Comment peut-on se fabriquer un light pen ? (je crois d'ailleurs que vous en aviez parlé dans le numéro 1 de la Commode)

- Comment peut-on se fabriquer des paddles, car ceux qui sont en vente à Procep ne conviennent pas à mon budget financier.

- En ce qui concerne l'article traitant sur les différentes manipulations à réaliser avec les magnétophones, je voudrais savoir s'il serait possible de relier deux CBM à distance au moyen d'un modem. Si oui, pourrait-on avoir le schéma de ce modem, ou alors en existe-t-il de commercialisés par Procep ?

- Est-ce-qu'il existe une carte spéciale, qui permet de transformer le petit écran du CBM 3032 40 colonnes en 80 colonnes, car il est assez ardu d'adapter des dessins ou des tableaux de comptes de l'écran à l'imprimante ?

- Pourrions-nous avoir un banc d'essai de l'interface haute résolution du CBM 3032, avec toutes les caractéristiques techniques ?

- Est-il possible, en rajoutant un second écran au CBM, de réaliser un programme : en langage machine, qui permettrait de générer la couleur, si l'écran rajouté est un écran couleur ?

- Existe-t-il une interface sur le CBM, permettant de rajouter des emplacements pour mettre des éproms en plus ?

- Peut-on transformer notre CBM 32K, en CBM 64 K, à l'aide d'une ROM multiplexeur.

Voilà ! je crois que cela est tout !

En espérant que cette lettre soit publiée avec les réponses, de façon à ce que tout le monde en profite,

Je vous prie d'agréer, l'expression de mes sentiments CBM-Istes les meilleurs.

Eric DEPAYE
06700 ST LAURENT DU VAR

L'unité CBM2031 est en vente depuis quelque temps au prix de 3950 F HT.

Pour un certain nombre de questions voir dans ce numéro.

Il n'est pas possible de transformer un 3032 petit écran en 80 colonnes; cela imposerait trop de modifications internes, comme d'ailleurs de faire passer la mémoire à 64K.

On ne peut pas brancher directement un écran couleur sur la sortie vidéo du CBM. Pour faire de la couleur, il faut lui ajouter une interface trés onéreuse de génération de la couleur.

Pour l'échange des programmes, il est nécessaire que l'auteur de l'article soit propriétaire du programme.

*
* *

Votre second numéro s'est étoffé, il est mieux présenté et le lecteur s'en réjouira.

Il ne nous reste qu'à espérer une parution dans les délais ! Merci d'avance.

Sur un plan général, je suis persuadé que le domaine privilégié de la micro-informatique et, en particulier, du matériel Commodore, est celui des applications professionnelles.

En effet :

- La principale caractéristique de l'ordinateur est sa possibilité d'effectuer inlassablement une tâche répétitive dans des délais très courts : calculs scientifiques, tri de fichiers, classements stockage de données.

- On se lasse finalement assez vite des jeux vidéo, la gestion familiale est rarement si complexe qu'elle justifie l'usage d'une machine autre qu'une simple calculatrice à mémoires (n'oublions pas le temps de mise en oeuvre d'un programme, non négligeable, même sur un micro) et les joies de la programmation n'on qu'un temps : pour un néophyte, la mise au point est terriblement longue et l'on ne fera jamais aussi bien qu'un programme écrit par des professionnels, partiellement ou totalement en langage machine (ex.: Visicalc, traitement de textes, paye, compta, gestion de fichiers...).

- Enfin, même à des prix relativement abordables, l'investissement de 10.000 à 30.000 francs minimum, ne se justifie pas pour un simple usage domestique (on se rend très vite compte qu'il est indispensable de posséder une imprimante et 2 floppys).

Pour ma part, j'ai vraiment tout essayé avec mon ensemble Commodore 3001 : j'ignorais tout de l'informatique et cela fut pour moi une véritable révélation qui s'est transformée petit à petit en passion. Mais que de nuits blanches pour des mises au point de quelques lignes de programmes !

Combien d'erreurs dont la correction entraîne d'autres erreurs encore plus insidieuses ! Il est vrai qu'avec de la méthode et un fignolage de l'organigramme... Mais, c'est ennuyeux de faire un organigramme et, après tout, on n'a pas acheté un ordinateur pour faire des pages d'écriture avant de s'en servir !

Quant aux programmes publiés dans les (autres) revues spécialisées, si on arrive à les lire, ils sont généralement truffés d'erreurs !

Et puis, la finalité de la technique, n'est-elle pas d'être au service de l'Homme et non l'inverse ? Il n'est plus nécessaire d'être mécanicien pour conduire une voiture et tout le monde doit pouvoir utiliser un micro-ordinateur en ignorant dans quel langage est écrit le programme.

Malgré ces propos amers, je ne regrette rien, le domaine est passionnant et les résultats sont concrets : Après un an d'utilisation, j'ai de nombreuses applications qui, dans mon métier, me rendent d'inestimables services. Le gain de temps est moins certain, car la mise en place encore trop récente, mais certaines tâches à caractères fastidieux me sont épargnées et cela compte aussi.

Mes utilisations principales sont :
- La paye
- La comptabilité générale
- Les statistiques
- La facturation.

Pour ces utilisations, j'emploie le programme Paye mis au point car la société Lanceau à Besançon, le programme Comptabilité Générale Procep, l'excellent programme Visicalc, un programme de statistiques de ma fabrication, et

une adaptation du programme gestion de fichiers Procep pour l'établissement et le contrôle des factures.

Exerçant la profession d'expert en automobiles et risques divers pour le compte de compagnies d'assurances, j'ai lui avec beaucoup d'intérêt votre article consacré au lecteur de cartes Chatsworth MR 500. Malheureusement il n'existe pas d'importateur et, dans ce cas, comment approfondir ses possibilités d'adaptation à une utilisation spécifique ? Quels seraient, en outre, les recours en garantie possibles en cas d'incidents ?

S'il m'était possible d'obtenir un rendez-vous avec Monsieur Daniel-Jean DAVID, je vous en serais très reconnaissant. J'ajouterai, d'ailleurs, qu'il est l'auteur qui m'a sans doute le plus aidé dans mon approche de la micro-informatique tant ses libres et publications d'articles sont clairs et compétents.

Un dernier voeu : de plus en plus d'articles, et les plus "fouillés" possible.

Amicalement.

P.A. SAULOU
91450 SOISY / SEINE

*
* *

Je suis auteur d'un programme de Rubick's Cube en trois dimensions pour CBM 3/4000. A ce sujet, je vous remercie pour l'article spécial compatibilité qui m'a permis d'en programmer deux versions.

Je voudrais connaître la procédure employée par votre journal spécialisé pour recevoir ces programmes relativement long au point de vue listing. Secundo, faut-il accompagner ce programme (ici en langage machine) d'un article détaillant son contenu (méthode de programmation, astuces) ce qui pourrait constituer une initiation au langage utilisé, ou au contraire ne pas dévoiler (en tout cas, pas explicitement) le travail de l'auteur, mais fournir au lecteur une "notice" détaillée du programme. (ce qui irait dans le sens des défenseurs de la protection des logiciels). Mais, ne serait-ce pas instructif et constructif de pourvoir l'utilisateur d'une connaissance approfondie des logiciels parus dans une revue ? En effet, ne serait-ce pas inciter les lecteurs à programmer eux-mêmes, soutenus qu'ils seraient alors, par des astuces de programmation parfois difficiles à trouver ; sans leur emploi le programme est, soit trop long, soit ne fonctionne quelquefois jamais (très décourageant !), ce qui pousse au piratage, dénoncé justement par les auteurs, de logiciels tout frais et pimpant plutôt qu'au travail de "débuggage", très éprouvant pour les nerfs.

- Votre revue s'est nettement améliorée depuis le numéro deux, où l'on découvre déjà plus d''astuces' et de programmes. J'ai particulièrement apprécié le programme de Olivier Barthélémy, qui est à la fois culturel et 'informatif'. Je constate néanmoins que pour cet article c'est l'explication des commandes nécessaires à l'utilisateur qui à été choisis. L'auteur le dit clairement : "Voici la notice d'utilisation". L'étude du programme m'a montré que Olivier Barthélemy emploie toutes les fonctions mathématiques du CBM. L'astronomie y est pour beaucoup de choses !

Il utilise aussi, mais est-ce un avantage au point de vue vitesse d'exécution ? largement les sous-programmes, placés d'ailleurs au début ; je suppose pour améliorer les performances du programme en vitesse (cf. la Commode numéro 1).

J'ai constaté qu'avec l'écriture resserrée de l'imprimante, les caractères de contrôle ne sortait pas bien (était même illisible pour être déplaisant).

Ceci irait-il encore dans le sens de la protection du logiciel ? Ou plutôt voudriez-vous faire travailler les utilisateurs lors de

l'introduction du programme au clavier pour reprogrammer la présentation à l'écran ?

Et en conclusion, remplissez bien les tiroirs ajoutés à la Commode après la parution de la lettre de M.P.A. Saulon dans le numéro 2.

P. DEVALLOIS
92100 BOULOGNE

Merci de votre lettre.

Nous sommes tout à fait partisans de publier votre programme de RUBIK's CUBE. Nous pensons qu'il serait d'ailleurs excellent de publier aussi les explications de la programmation que vous avez faite : la Commode veut être un journal didactique permettant à ses lecteurs de progresser. Nous pensons que cela pourrait se faire sous la forme de deux articles :

- dans le premier, le programme et le mode d'emploi.

- dans le second, des explications sur la méthode de résolution du cube et sur la programmation.

Seule ombre au tableau : nous n'avons pas encore les moyens de payer nos auteurs (sauf, bien sûr, par des abonnements gratuits).

Alors, à bientôt.

*
* *

Votre initiative, de lancer une revue concernant un matériel précis, correspond certainement à un besoin. Pour certaines applications ou simplement par curiosité, il devient indispensable de savoir un maximum de détails comment fonctionne ou plutôt quelles sont les particularités du matériel que l'on a en face de soi.

Je regrette quelque peu que votre numéro 1 s'en tienne, sauf quelques exceptions à des généralités, mais je regrette encore plus d'être resté sur ma faim, n'ayant toujours pas reçu le numéro 2.

Si vos abonnements sont gérés par un système informatique, j'espère ne pas avoir été victime d'une défaillance de celui-ci !

J'espère surtour recevoir enfin ce numéro 2 (avant le numéro 3 si possible !) et parvenir à une connaissance de mon matériel qui puisse me permettre de vous envoyer parfois un article.

Par avance, merci, et mettez ces petites pointes sur le compte de l'intérêt porté à votre revue.

René CHAFFARD
13013 MARSEILLE

La Commode a d'abominables retards (qui espérons-le, seront résorbés), mais n'hésitez pas à vous plaindre en cas de non réception, car il peut aussi y avoir des problèmes postaux.

*
* *

Les deux premiers numéros de "La Commode" sont très intéressants pour les utilisateurs COMMODORE.

"LA COMMODE" peut et doit être une bourse d'idées et d'articles pour ceux-ci.

Je vous soumets mon problème :

Je possédais un PET 2001 dernière version (CBM 3000) et un ensemble de jeux, certains en BASIC, d'autres en langage machine ("envahisseurs", par exemple). J'ai revendu mon PET et acheté un CBM 4016 grand écran. Mes programmes BASIC fonctionnent bien sûr, mais pas ceux qui sont écrits en langage machine.

Connaissez-vous le moyen de faire tourner ces programmes sur le 4016, et si oui, pouvez-vous donner un exemple.

En vous remerciant de votre attention,
Je vous envoie toutes mes salutations.

J. CLEMENT
14470 COURSEULLES

Voir exemple dans le numéro 2. Mais si vous n'avez pas le texte source, il faut d'abord analyser le programme.

*
* *

Suite à votre article du numéro 2 page 35, "Comment augmenter la mémoire de votre 3008 ou 3016, je me permet de vous faire une petite remarque. Vous avez oublié un cas : le passage de 3008 à 3016.

Ce n'est pas volontairement que je ne veux que 16K RAM, mais ma carte a 3 trous (faits à la perceuse sans doute) dans la seconde rangée, ce qui rend l'extension 32 K RAM impossible.

Vous serait-il possible d'indiquer comment faire pour avoir 16K RAM, en remplaçant les 4108 par des 4116. (ce que je demande est donc le brochage des cavaliers à l'arrière de la carte) ?

D'autre part, dans mon cas, ne serait-il pas plus simple de faire une extension mémoire sur le connecteur bus.(comme sur les "anciens" PET) dans le genre de l'extension "Petite 24K".

A propos des disques CBM, il existe maintenant le monodisque CBM 2031 (171 K). Pourriez-vous, si vous le pouvez, donner quelques renseignements sur ce lecteur de disque ?

Dernière chose, comment se procurer le numéro 1 (mon abonnement commence au numéro 2)

En vous remerciant de l'attention que vous porterez à mes questions.

E. LEGENDRE
21121 DAIX

N.B. : Bravo, les listings (nombreux) sont très lisibles.

Pour passer du CBM3008 au 3016 il suffit d'ôter le cavalier E et de mettre le cavalier F. Par contre il est possible de passer quand même à 32K, même avec les trous! Pour cela il faut utiliser des supports à "wrapper" et rétablir les liaisons qui manquent. Bien sûr il est nécessaire de disposer du schéma du PET (dans "THE PET REVEALED" par exemple).

Le CBM2031 se comporte exactement comme la 3040/4040.

Pour se procurer le numéro 1, il suffit d'envoyer un chèque de 35F à La Commode.

*
* *

Voilà bientot un mois que je me suis abonné à votre revue, et j'ai déjà dévoré tous les articles des numéros 1 et 2. Je les trouve très intéressants et passionnants. J'ai essayé et analysé tous les programmes que vous avez fait paraître. Aussi, je me permettrais quelques remarques :
- Dans le programme Ephémérides du numéro 2 page 56 ligne 62, il faut remplacer X par T dans INPUT.
- Dans le programme Librarian du numéro 2 page 19 et sans doute 20, lignes 33 et 34, il y a des erreurs de POKE. La valeur 1281 tombe en plein dans la ligne 10, alors qu'il faudrait qu'elle tombe dans la ligne 11. Aussi, je propose POKE (1294 + I) ligne 33 et POKE 1300 ligne 34.
- Dans le programme Cross référence du numéro 2 page 22, ligne 510, quel est le sens du test :
IF A$ = Q $?
- Dans l'article de l'utilisation de PRINT du numéro 2 page 13 et 14, j'ai remarqué que vous avez passé sous silence l'emploi suivant :
POKE A,L : POKE B,C : SYS D : PRINT...
avec 0<=L<=24 0<=C<=39

A = 216 pour 3000/4000,
245 pour 2001
B = 198 pour 3000/4000,
226 pour 2001
D = 57471 pour 4000,
57949 pour 3000,
58843 pour 2001.

D'autre part, je possède un système 4016-4040-4022. Pourriez-vous me dire pourquoi le 4016 n'est pas extensible en 4032, et s'il n'y aurait pas un moyen de tourner la

difficulté ? Je serais intéressé également par un article sur les fichiers USR sur 4040. Etant enseignant, je voudrais protéger mes programmes pour que les élèves n'aient pas accès aux solutions des questions auxquelles ils doivent répondre. Comment dois-je faire ?

Pour terminer, je vous soumets une bizarrerie que je n'ai pas pu élucider :

```
10 X = 1/6
20 Y = 1/2 - 1/3
30 IFX = Y THEN PRINT "EXACT"
```

Ce programme marche.

```
10 X = - 1/6
20 Y = 1/3 - 1/2
30 IFX = Y THEN PRINT "EXACT"
```

Ce programme ne marche pas.

J'attends impatiemment la parution du numéro 3, et vous encourage vivement à continuer de nous dévoiler tous les attraits des CBM.

Thierry GELY
75013 PARIS

Il ne faut jamais faire de tests par égalité avec les nombres réels.

Merci de signaler l'erratum de LIBRARIAN, maintenant corrigé.

Le 4016 est extensible en 4032, voir numéro 3 page 36.

*
* *

Ayant eu connaissance récemment du numéro 2 de La Commode, j'y ai découvert, p.31 (article de D.J. DAVID sur les unités de disques), au sujet de la compatiblilité BASIC 2.0 / BASIC 3.0 des renseignements inestimables pour moi, possesseur d'un 3032 raccordé à une 8050, et que je n'avais jamais vus ailleurs, (peut-être ai-je mal lu les notices CBM/PROCEP ?).

Aussi, je m'empresse de vous envoyer le montant de l'abonnement aux 4 prochains numéros avec mes remerciements !

Je suis intéressé par les liaisons par modems... J'aimerais voir un article sur les possibilitées actuelles, à partir du matériel CBM.

Cette lettre a été rédigée par mon mini-traitement-de-texte personnel: j'ose espérer que vous apprécierez la signature!

REMBOURSEMENT D'UN PRET (P.15)

Je souhaiterais voir un complément à ce très intéressant article, pour résoudre le problème pratique suivant :

Emprunt d'une somme S sur N années, taux annuel T, avec remboursement par annuités constantes.

- calcul des annuités
- calcul, dans chaque annuité, de la part remboursement et de la part intérêts ?

CORRESPONDANCE DES POINTS D'ENTREE BASIC 1/2/4

BRAVO ! Mais :
FFD2 : SORT UN CARACTERE...Sous entendu : dont le code ASCII a été mis dans (A)
Comment le deviner ?
FFC9 : ETABLIT UN PERIPHERIQUE DE SORTIE :... Sous entendu : dont le No a été mis dans (X) (...sauf erreur !)
Comment le deviner ?

EPHEMERIDES ASTRONOMIQUES (P.58 2ème colonne)

La latitude de PARIS n'est sûrement pas +180 50' 24" ?

Il manque à ce magnifique programme, le calcul du temps sidéral local à l'heure de l'observation, nécessaire au réglage de l'équatorial...

COMPATIBILITE BASIC 2.0 BASIC 4.0 (P.31)

Qui aura deviné que, pour que ça marche, il faut avoir préalablement tapé : OPEN 80,8,15 ?

Pitié pour les débutants !!!
OPEN82,8,2,"0:nom",L,"+CHR$(taille) qui comporte un nombre impair de", ne marche sûrement pas ! Un exemple serait le bienvenu !

1) En réponse à votre lettre du 21/04/82, je ne vois pas d'inconvénient à ce qui vous fassiez état des termes de ma lettre du 19/04/82 dans votre plaquette de

présentation.

2) Suite à la PARUTION DE LA COMMODE No 3 :

2.1 P.50 : Merci d'avoir publié l'erratum que je vous avais signalé.

2.2. P.21 : Vol d'oiseau :

Je ne résiste pas au plaisir de vous communiquer une variante de ce programme, qui produit exactement le même effet, avec une certaine économie de moyens : (un seul type de mouvement curseur, pas de concaténation, utilisation de TAB(I)) : cela peut intéresser certains lecteurs.

```
100 REM J.PIERRAT VERSAILLES
110 REM [VOL D'OISEAU]
120 REM REF. LA COMMODE NO.3
130 REM
200 PRINT"♥"
210 FORI=1TO17
220 PRINTTAB(I) " ╭●╮"
230 FORK=1TO100:NEXT
240 PRINTTAB(I)"▯╰●╯"
250 FORK=1TO100:NEXT
260 PRINTTAB(I)"▯    "
270 NEXT
280 FORI=1TO200:NEXT
290 GOTO200
```

2.3. P.45 : Bibliographie :

Concernant le B.C.M. et la pratique du PET/CBM No 2, les remarques de J.P. LAURENT recoupent si exactement les miennes, que je n'ai rien à ajouter.

Concernant la pratique du PET/CBM No 1, je me permets de vous transmettre quelques observations personnelles.

3) Je compte vous faire parvenir ultérieurement un article sur une version simplifiée, mais performante, de mon TRAIDATA , utilisé pour produire cette lettre.

Cordiales salutations :

J. PIERRAT
78000 VERSAILLES

Merci. Les remarques sur la Pratique du PET/CBM vol 1 sont en rubrique bibliographie.

*
* *

Chers Membres de la Commode

Je vous écris afin de vous féliciter pour votre excellente revue qui m'a enthousiasmé (et mon VIC aussi).

Elle est sérieuse, bourrée d'astuces et de programmes.

J'espère que vous continuerez à publier des programmes pour le VIC (jeux et applications sérieuses). Je possède le VIC depuis 1 mois et demi, et j'essaye de créer quelques programmes (comptez sur moi pour vous les envoyer).

Par ailleurs, je vous joint un chèque afin de recevoir les deux premiers numéros de la Commode, que j'espère recevoir rapidement.

Vous serez-t-il aussi, possible de me donner l'adresse de la revue "VIC Computing" ainsi que les modalités d'abonnement.

Si vous connaissez d'autres revues concernant le VIC, n'hésitez pas à me les envoyer.

En vous félicitant à nouveau,
Amicalement

Frédéric GERSON
31500 TOULOUSE

VIC-COMPUTING 39-41 NORTH ROAD
LONDON N7 9DP
Europe = 9 livres (1 an)

*
* *

Une cassette achetée dans le commerce est écrite en langage machine (un jeu). On ne peut ni lister, ni sauver, ni imprimer ce programme. Y-a-t-il une astuce ?

Jacques VAISSIERE
03340 NEUILLY LE REAL

Utiliser le moniteur L.M. à condition que le programme ne soit pas AUTORUN.

*
* *

Premières impressions sur le CBM 9000

Un système bien allèchant !

Le modèle -pour le moment- le plus élevé de la gamme Commodore est arrivé en quelques exemplaires en France et nous avons eu la chance de le prendre en mains.

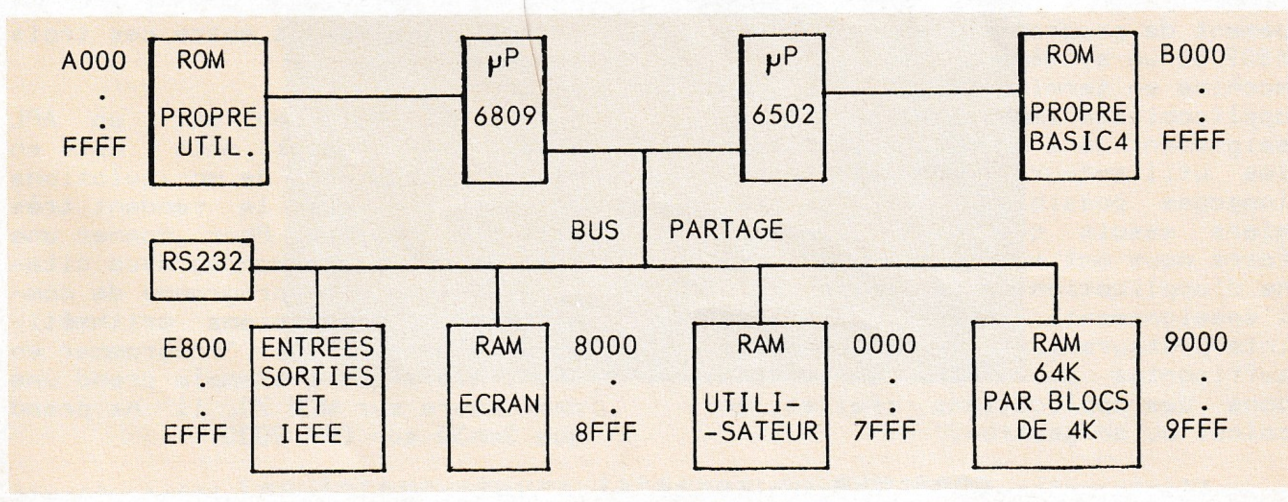
Le nom de ce modèle? Commodore l'appelle CBM 9000 et ajoute, comme surnom, soit MICROMAINFRAME (c'est à dire micro-gros ordinateur, ce qui est un peu contradictoire, mais décrit assez bien le niveau de performances auxquelles il faut s'attendre), soit SUPERPET, mais ils avaient appelé ainsi tous les modèles postérieurs au PET.

Ne sachant quelle sera l'appellation définitive en France (Super CBM conviendrait assez, mais il y aura toujours dans l'avenir des modèles encore plus "super"), nous dirons CBM 9000.

Extérieurement, le CBM 9000 se présente exactement(à part l'inscription de la façade) comme un CBM 8032. La seule différence est qu'on trouve à droite un jeu de deux interrupteurs à trois positions: pour le premier (I1) les trois positions sont: 6502,6809 et PROG; pour le second (I2), les trois positions sont: READ/WRITE, READ ONLY et PROG.

C'est tout simplement que le 9000 n'est autre qu'un 8032 qui a en plus:
1- un microprocesseur 6809 avec ses ROM propres
2- une carte RAM 64K.

Plus précisément, la configuration correspond à la figure ci-dessous. D'un côté vous avez le



Configuration du CBM 9000

6502 avec ses ROM propres (le BASIC 4.0 de Commodore de B000 à FFFF et un emplacement libre en A000 pour EDEX par exemple). D'autre part vous avez le 6809 et ses ROM propres de A000 à FFFF, contenant des sous-programmes utilitaires faisant partie du logiciel Waterloo.

Lorsque Il est en position 6502, vous avez un CBM 8032 parfait, dont vous pouvez utiliser tous les logiciels comme OZZ ou TRAITEXT. La seule différence est que vous disposez en plus de 9000 à 9FFF d'une RAM de 64 K (à un instant donné seul un bloc de 4K est actif) que vous pouvez utiliser à votre gré, en tenant compte de sa pagination.

ATTENTION: vous n'avez pas un 8096 en ce sens que la pagination n'étant pas faite comme dans le 8096, les logiciels préparés pour le 8096 ne seront pas compatibles, mais il est clair que des versions compatibles 9000 seront développées.

Lorsque Il est en position 6809, vous disposez d'une machine nouvelle! Cette machine nouvelle est livrée avec un ensemble de logiciels appelés les micrologiciels WATERLOO (université Canadienne) formés principalement d'interpréteurs des langages BASIC, FORTRAN, PASCAL et APL (plus COBOL dans un avenir proche). Il y a en outre un éditeur, un système de développement de programmes en assembleur 6809 et un système de gestion de la machine en terminal RS232. D'où une applicabilité immédiate dans l'enseignement où l'on souhaite former les utilisateurs aux différents langages possibles. Mais les premiers essais que nous avons pu faire nous ont montré que le domaine d'application ne se limite pas à l'enseignement. Les différents interpréteurs sont de performances suffisantes pour être employés dans des applications réelles, en calcul ou en gestion.

L'interpréteur BASIC Waterloo est plus lent que le BASIC Commodore (dont on dispose de toutes façons sur le CBM 9000), mais il reste honorable et il a des extensions qui méritent qu'on s'y arrête:

- d'abord, toutes les structures de la programmation structurée

```
IF THEN habituel, mais aussi
IF THEN..ELSEIF..ELSE..ENDIF
LOOP..ENDLOOP
LOOP..UNTIL
WHILE..ENDLOOP
LOOP..QUIT..ENDLOOP
```

- les instructions sur matrices: somme,produit etc...

- la définition de fonctions sur plusieurs lignes

- des procédures appelables par nom et avec transmission de paramètres

- des identificateurs de 31 caractères tous significatifs

- des facilités d'édition dépassant EDEX. Les valeurs des variables sont conservées même si on a modifié le programme.

Le FORTRAN contient à peu près tout FORTRAN IV, mais il a reçu des extensions de programmation structurée et de gestion des chaînes de caractères qui dépassent de loin FORTRAN 77.

Le PASCAL est un PASCAL Zurich complet. Les extensions structurées de BASIC et FORTRAN créent un certain nivellement entre ces trois langages.

L'APL Waterloo est un APL absolument complet mais doté en plus d'extensions de manipulations de fichiers qui le rendent très apte à la gestion. Pour donner une idée de ses performances nous citerons l'exemple de programme de compilation d'expressions arithmétiques de notre livre "Programmer en APL". Alors que l'exemple prend une demi heure sur MCM 70, il ne prend que 2mn30 sur CBM 9000!

Chacun des interpréteurs Waterloo se charge à partir du disque dans la zone de mémoire supplémentaire de 64K. On peut ensuite, en manoeuvrant l'inter-

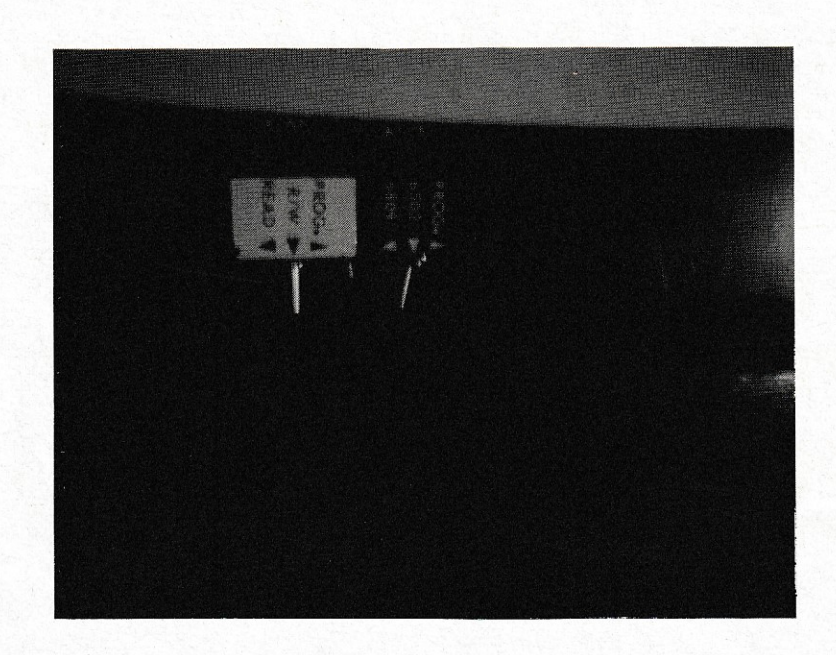
rupteur il mettra cette mémoire en "READ ONLY" ce qui évitera toute altération de l'interpréteur par une fausse manoeuvre, comme s'il était en ROM.

La zone de mémoire vive habituelle des CBM (de 0000 à 7FFF) est, elle, disponible pour le programme utilisateur et ses variables. Il y a donc 32K de disponibles, moins les variables système.

Au niveau périphériques, à part l'interface RS232 dont il est doté en supplément, le CBM 9000 admet exactement tous les périphériques de la gamme CBM. En mode 6809, le 6809 prend en charge la gestion du bus IEEE donc tous les périphériques habituels sont accessibles aux logiciels Waterloo.
En ce qui concerne les disques, comme le DOS est dans l'unité de disques, les fichiers constitués tant sous le contrôle du 6502 que du 6809 sont totalement compatibles et compatibles avec les
créés par un CBM 4000 ou 8000.

En fait le CBM 9000 résulte d'un accord très astucieux entre l'université de Waterloo et Commodore: on avait besoin pour les logiciels Waterloo d'un clavier-écran et disque bon marché; le CBM 8000 offre tout cela pour former un système impressionnant, cumulant les avantages du CBM avec les langages BASIC 4.0, COMAL, FORTH, LISP, PILOT, Assembleur 6502 et ses programmes d'application OZZ, TRAITEXT, VISICALC etc... avec les avantages des logiciels Waterloo: Assembleur 6809, BASIC structuré, FORTRAN, PASCAL, APL et bientôt COBOL. On a au total une belle palette de langages: ce serait bien le diable s'il n'y en avait pas un qui convienne à votre application.

Daniel-Jean DAVID



Les interrupteurs latéraux du CBM 9000.

Mieux utiliser BASIC

Les principaux outils d'une bonne programmation

Malgré l'apparition sur micro-ordinateurs de langages structurés comme le Pascal, le Basic continue a être le plus utilisé sur les différents appareils à la disposition des utilisateurs.

Malgré les attaques nombreuses qu'il a subies, le Basic reste un bon langage de programmation de haut niveau, surtout avec toutes les améliorations qui lui ont été apportées.

Mais, un certain nombre de bonnes habitudes doivent être prises pour éviter de produire des programmes illisibles et presque impossibles à maintenir.

Les commentaires

Ils sont indispensables pour éviter de longues interrogations au moment de la maintenance surtout si elle doit être faite par une autre personne que l'auteur du programme. Et même pour ce dernier, ils sont très utiles comme rafraîchissement de mémoire.

Toutefois et surtout en micro informatique, ces commentaires ont deux graves inconvénients :

- Ils tiennent beaucoup de place en mémoire.
- Ils ralentissent les opérations.

Pour surmonter cela, il y a intérêt à posséder deux états du programme.

- un état "archives" riche en commentaires très apparents en listing.
- un état "opérationnel" où tous les commentaires sont supprimés (ce qui peut être fait par programme) et où les lignes sont numérotées de 1 en 1 à partir de un (pour cela un programme de renumérotation est indispensable)

A chaque maintenance effectuée à partir du programme "archives" un nouveau programme "opérationnel" est édité.

On obtient ainsi les performances optimales du micro-ordinateur.

Organisation du programme

Normalement, pour avoir un maximum de clarté et de structuration du programme, celui-ci devrait être systématiquement divisé en deux groupes :

Le programme principal
Les sous-programmes effectuant chacun une action précise et limitée.

Le programme principal ne servant que de lien chronologique à

l'appel du sous-programme.

Toutefois, pour des questions de rapidité d'exécution et d'encombrement mémoire, on est en général appelé à un compromis. On ne met alors en sous-programme que les actions qui sont suffisamment appelées ou certaines initialisations de paramètres devant être appelées de différents points d'entrées du programme principal.

Où placer les sous-programmes

Il y a en général avantage à les placer en tête de programme pour gagner du temps.

```
0001 GOTO entrée principale
XXXX Sous routine
.
.
.
YYYY entrée principale
```

En effet, la recherche d'une adresse se fait toujours à partir du début du listing.

Toutefois, pour des raisons pratiques, les sous-programmes obtenus à partir de bibliothèques de programmes seront ajoutés en fin de programme (règle d'emploi de l'ordre "append"). Mais on peut par la renumérotation sélective les placer aussi en tête.

Comment lier les sous-programmes au programme principal

Lorsqu'un sous-programme doit manipuler des variables numériques ou alphanumériques, il est très important d'éviter ce que l'on appelle les "effets de bord", c'est-à-dire, les modifications intempestives des variables du programme principal ou d'un autre sous-programme. Pour cela, il convient que le sous-programme utilise des variables réservées avec échanges en entrée et en sortie avec les variables du programme principal.

```
ENTREE S.Pg      Var.S.Pg =Var Pg
Processus
SORTIE S.Pg      Var Pg. = Var.S.Pg
```

Exemple : Indice de boucle

Si dans un sous-programme, on utilise l'indice I déjà en action dans le programme appelant, il y aura perturbation.

```
FOR I = 1 TO N
GOSUB Spg          For I = 1 TO M
...                ...
...                NEXT
...                RETURN
NEXT
```

On exploitera en sous-programme un indice I0, par exemple.

Le seul cas où il ne faut pas faire de séparation, c'est quand la sous routine doit servir à initialiser des paramètres.

Taille des sous-programmes

Dans l'idéal, chaque sous-programme devrait effectuer une seule action bien précise.

Toutefois, là aussi, il faut trouver un compromis avec les nécessités de rapidité et de taille mémoire. On essaie d'obtenir l'équilibre optimal entre ces obligations contradictoires en gardant à l'esprit que plus une unité de programme est courte, plus elle est facile à dépanner et à maintenir.

Paramétrisation

Un programme est une chose vivante qui doit continuellement être remaniée. Prenons l'exemple d'une date où le jour seul est à rentrer, le mois et l'année étant affichés par l'appareil. Il est évident que si la date apparaît en clair dans des lignes de programme très dispersées, on perd du temps et risque des oublis à la maintenance. Par contre, si MM$ représente le mois et AA$ l'année, une seule mise à jour est nécessaire.

La règle est très générale. Toute donnée répétitive doit être paramétrée. Si, par exemple le mot prix doit être répété plusieurs fois, on posera :

PX$ = "PRIX"

de même pour le diviseur 256, on pourra écrire :

DI = 256

Cette façon de faire a aussi l'avantage d'accélérer les opérations, la donnée devenant accessible immédiatement sans changement de format dans l'unité centrale. Il convient donc d'écrire, soit en début de programme, soit dans une routine spéciale, un véritable dictionnaire des paramètres utiles du programme.

Les principaux paramètres utiles

A - paramètres de gestion d'écran

Il est intéressant de disposer des chaînes de caractères suivantes :

1) Vecteur espace (ou blanc)
BL$ = "40 espaces"

2) Vecteur arrière
AR$ = "40 curseur gauche"

3) Vecteur avant
AV$ = "40 curseur droite"

4) Vecteur descente à partir de HOME
DE$ = "HOME 24 curseur bas"

5) Vecteur montée
MT$ = "24 curseur haut"

A partir de ces vecteurs, par concaténation et traitements de chaînes, on peut en constituer beaucoup d'autres. Comme un vecteur effacement par exemple,

EF$ = LEFT$(BL$,K)+LEFT$(AR$,K)

(K est la longueur d'effacement)

et pouvoir sans risque de surcharge afficher une donnée sur une ligne déjà écrite.

Selon les besoins, on peut ajouter :
PT$ = "......." pointillés
SL$ = "-------" pour souligner

Et bien d'autres que nous laissons à l'imagination du lecteur.

B - Paramètres de gestion de périphérique.

Ils sont très utiles pour appeler au choix par programme un périphérique.

On a intérêt à créer :
PP = numéro du périphérique
NF = numéro de FILE
AS = adresse secondaire

et pour les lecteurs de disquettes:

DO$ = "0:" D1$ = "1:" EO$ = "@0:"
E1$ = "@1:" SR$ = ",S,R"
SW$ = ",S,W" CR$ = CHR$(13)

Dans cet esprit, chacun pourra créer les paramètres de son choix, selon les périphériques utilisés. Certains paramètres peuvent être utiles à la fois pour différents périphériques comme ceux qui peuvent servir à l'écran et à l'imprimante.

C - Paramètres de calcul

Les constantes qui interviennent dans plusieurs calculs doivent être paramétrées, surtout pour des raisons de place mémoire et de vitesse d'exécution. En cas de modification, cela facilite également la maintenance.

Paramètres "Chaîne de caractère"

Les "mots" qui interviennent souvent doivent être paramétrés.

Exemple : si l'on demande souvent une validation, on peut avoir un mot "accord".

AC$ = "-> O/N"
O/N pouvant être écrit en vidéo inversée.

SIMULATION DE CERTAINS ORDRES DE PROGRAMMATION STRUCTUREE

IF.. THEN.. ELSE

Beaucoup de BASIC n'ont pas

cet ordre. Voyons comment le remplacer.

1er cas : affectation d'une valeur

```
IF A>0 THEN SG = 1 ELSE SG = -1
```

peut être remplacée par

```
100 SG = -1
110 IF A>0 THEN SG = 1
```

qui marche aussi bien et est peut être même plus logique.

2ème cas : saut à une ligne de programme

```
IF A>0 THEN GOTO1000 ELSE GOTO2000
```

peut être remplacé par

```
100 IF A>0 THEN 102
101 GOTO 2000
102 GOTO 1000
```

ou par :

```
100 IF A>0 THEN 1000
101 GOTO 2000
```

meilleur en général.

DO(action) UNTIL(condition vraie)

```
DO
A = A + 1
UNTIL A = N
CONTINUE
```

donne en BASIC

```
100 A = A + 1
110 IF A<>N  THEN 100
120 ...
```

DO WHILE (condition) ACTION

Donne en BASIC

```
100 IF A = N THEN 130
110 A = A + 1
120 GOTO 100
130 ...
```

On voit qu'ici, il est plus simple de vérifier la condition contraire (A = N est le contraire de A<>N).

Emploi de la Boucle FOR

```
FOR I = début TO fin : Action:NEXT
```

Il faut l'employer avec beaucoup de circonspection et uniquement lorsque le nombre d'actions est connu. Il est possible, mais toujours risqué de sortir d'une boucle avant qu'elle ne soit terminée.

Les causes principales d'erreurs sont :

- cas où début = fin. En effet, la boucle est quand même exécutée une fois.

Exemple

```
FOR I = 1 TO 1
A = A + 1
NEXT
```

A est incrémenté, ce qui est illogique.

-interférence d'indice avec un sous-programme, nous avons déjà évoqué le phénomène.

- boucles se chevauchant :

```
FOR I = A TO B
...
FOR J = C TO D
...
NEXT I
...
NEXT J
```

AVANTAGE MAJEUR DE LA BOUCLE FOR NEXT.

Sa rapidité. Sur CBM, nous avons obtenu pour la même action (incrémenter 1000 fois un nombre).

4"1 pour FOR...NEXT
13"3 pour DO...UNTIL
13"7 pour DO...WHILE

Notons que si on précise NEXT I il est perdu un peu plus de temps (0,3").

Précaution indispensable : toujours test avec DEBUT inférieur à FIN.

ENTREE DE DONNEES

PENDANT LE DEROULEMENT DU PROGRAMME

Cela est réalisé, soit par GET, soit par INPUT.

GET : entre les données, caractère par caractère, ou chiffre par chiffre.

Avantage : On peut exercer un contrôle très précis sur l'entrée de la donnée.

Inconvénient : L'entrée est ralentie et la programmation plus complexe.

INPUT : entre les données en bloc, mais possède de nombreux risques.

L'INPUT non protégé permet toutes les erreurs. De plus, un "RETURN" fait sortir du programme. Il faut en proscrire l'emploi.

Il y a plusieurs façons de protéger, dont nous ne donnons que quelques exemples, laissant chacun en imaginer d'autres.

1er exemple :

```
100 INPUT "réponse⇒ ⇒ * ⇐ ⇐ ⇐";RP$
    :IF RP$="*" THEN PRINT"⇑⇑"
    :GOTO 100
```

qui oblige à une réponse. Toutefois, reste possible l'erreur d'effacer l'étoile.

2ème exemple :

```
100  INPUT"réponse⇒ ⇒ ... ⇐ ⇐ ⇐ ⇐
     ⇐"; RP$:
     IF LEN(RP$)<>3 THEN
     PRINT"⇑⇑":
     GOTO 100
```

ici, il faudra une réponse en 3 caractères.

La protection peut être poussée aussi loin que l'on veut. Par exemple, on peut à la fois vérifier le nombre de caractères et leur nature. (numérique, alphabétique, signes grammaticaux...)

Exemple :

```
100 INPUT"réponse⇒ ⇒ * ⇐ ⇐ ⇐";RP$:
    IF RP$="*" THEN PRINT"⇑⇑" :
    GOTO 100
110 GOSUB 1000 : IF ER=1 THEN
    PRINT"⇑⇑": GOTO 100
120 ...
    ...
1000 FOR I1=1 TO LEN(RP$):
     AS=ASC(MID$(RP$,I1)):
     IF AS<48 OR AS>57 THEN
     ER=1 : RETURN
1005 NEXT
1010 ER=0 : RETURN
```

Le contrôle peut être raffiné jusqu'à vérifier la nature du caractère selon la place occupée. En paramétrant la routine précédente, on peut l'utiliser pour tous les INPUT du programme.

```
IF AS<L1 OR AS>L2 THEN ER=1:RETURN
```

Si par exemple, L1=65, L2=81, on vérifie que tous les caractères sont des lettres.

Exemple : Entrée d'une date

```
100 INPUT"DATE= JJMMAA =";DT$:
    IF LEN(DT$)<>6 THEN
    PRINT"⇑⇑" : GOTO 100
110 XX$=DT$:L1=48:L2=57:GOSUB1000
120 ...
    ...
1000 FOR I1=1 TO LEN(XX$):
     AS=ASC(MID$(XX$,I1)):
     IF AS<L1 OR AS>L2 THEN ER=1:
     RETURN
1005 NEXT
1010 RETURN
```

Le contrôle peut être affiné pour la date en vérifiant que MM est compris entre 1 et 12, JJ entre 1 et J(MM) où J(MM) est un tableau du nombre de jours de chaque mois corrigé pour MM=2, en tenant compte des années bissextiles.

Enfin AA peut être aussi contrôlé.

On peut raffiner en tolérant des entrées du type :

DT$ = "JJ.MM.AA"

par une analyse du type :

```
110 LN=LEN(DT$) :
    IF LN=7 OR LN=8 THEN GOSUB2000
    ...
    ...
2000 FOR I2= 1 TO LN :
     MI$=MID$(DT$,I2,1) :
     MI=ASC(MI$)
2005 IF MI>47 AND MI<58 THEN
     Y$=Y$+MI$
2010 IF MI<>46 THEN ER=1:RETURN
2015 NEXT
2020 IF LEN(Y$)<>6 THEN ER=1 :
     RETURN
2025 DT$=Y$
2030 RETURN
```

Et si ER=1 on fait recommencer l'INPUT.

TOUTEFOIS, on ne doit jamais oublier qu'il y a un compromis à adopter entre les risques d'erreurs et le temps perdu par les contrôles.

La bonne règle est d'ajuster le style de la saisie à la compétence de celui ou de celle qui aura à la faire. Cela étant valable pour tous les messages affichés pendant la saisie.

SIGNALONS pour finir que les caractères interdits sur un INPUT comme :

espace en tête ou en queue
virgule ou deux points
mouvements de curseur

peuvent être entrés à condition de placer des guillemets.

Ex. : INPUT A$
?"Jules, Albert" rentrera bien la virgule.

Il y a de plus pour les contrôles en général avantage à préférer les INPUT de chaîne de caractères par rapport aux INPUT numériques.

Types de messages utiles pendant la saisie

Disons tout de suite que dans un programme à usage professionnel les formules de politesse sont grotesques et résultent de manies anthropomorphiques. L'ordinateur n'est qu'un outil, n'en faisons pas un singe. On peut à la rigueur afficher des messages amusants pour réveiller celui qui fait une longue saisie.

Par contre, beaucoup de messages sont utiles ou même indispensables. Ces messages concernent :

1) La nature de la réponse
Chiffre ou lettre
longueur
acceptabilité
codée ou non codée
possibilité d'appeler une table de renseignements.
...

2) Le déroulement des opérations
Dans l'unité centrale :
"Je calcule attendez, mais arrêtez le programme si temps supérieur à 10' ".

Avec les périphériques
-Affichage en haut d'écran du périphérique en usage, des fichiers ouverts, des numéros de platine.
-Affichage de messages d'attente du même style que plus haut.
-Du rappel en fin de travail de la nécessité de faire certaines copies et comment les faire.

VERIFICATION DU PROGRAMME

Il est indispensable de vérifier qu'en dehors des fautes de syntaxe signalées en général par l'ordinateur, il n'y a pas de fautes logiques. Ce qui est loin d'être évident et demande une très grande attention.

Il faut pour cela faire tourner le programme "à la main" et voir si les relations logiques entre les variables se maintiennent jusqu'à l'impression du résultat (impression étant pris au sens large de tous supports utilisés : papier, bandes magnétiques, disquettes, etc.)

C'est à ce stade de vérification que le découpage judicieux en

sous-programme s'avère très puissant pour faciliter le suivi de la logique.

Au contraire, un programme embrouillé et plein de "GOTO" est presque impossible à vérifier par cette méthode, et l'on doit se contenter de "jeux d'essais". Les jeux d'essais qui consistent à mener de front sur ordinateur et à la main l'exécution d'une action dans un certain nombre de cas pré-établis, ne sont que des palliatifs qui risquent de laisser passer certains cas exceptionnels et oublis.

CONCLUSION

Le langage BASIC qui a le très gros avantage de pouvoir très vite être enseigné à des profanes, subit néanmoins les assauts répétés des "structuralistes".Nous pensons qu'il ne faut pas montrer trop de sectarisme dans ce domaine, et que le choix doit dépendre de l'environnement des problèmes à résoudre.

Le BASIC a reçu de nombreux perfectionnements. De plus, il existe maintenant, à côté des interpréteurs, des compilateurs permettant d'obtenir des programmes objet performant.

A condition de s'astreindre à une certaine discipline, comme de limiter la taille des unités de sous-programmes et de ne garder que les "GOTO" nécessaires, on peut obtenir d'excellents programmes BASIC faciles à maintenir.

D'autre part, d'un point de vue intellectuel, ils sont souvent plus amusants à construire, car opérant dans un carcan beaucoup moins rigide et automatique que celui de la programmation structurée, ils font plus appel à l'imagination.

Disons pour en terminer : que chacun demeure libre selon les circonstances de faire son choix, et "paix sur la Terre aux programmeurs de bonne volonté".

Jean DELAVILLE

PETits trucs et COMBines

Voici un exemple d'utilisation très simple des routines système du CBM 3032 : la conversion décimal-hexadécimal.

a) Partie machine (6 octets à rentrer par le moniteur langage machine)

```
027A A9 00      LDA  # N
027C 20 75 E7   JSR  $E775
027F 60         RETURN
```

b) Partie BASIC

```
100 INPUT "DECIMAL <256";N
110 POKE 635,N : SYS 634
```

Jean DELAVILLE

RANDOM en BASIC 4.0

"Hasard" ne signifie pas "n'importe quoi".

En Basic 4.0 pour avoir une séquence pseudo-aléatoire qui soit différente à chaque mise sous tension du CBM il faut utiliser RND(0). Les nombres au hasard sont fournis par une "FREE RUNNING CLOCK" (je ne connais pas le terme technique français).

En effet, en Basic 4.0 (à la différence des précédents Basics de Commodore), RND(X) quand X X 0 génère toujours la même séquence quelle que soit la valeur de X. Il devient donc inutile et même nuisible d'utiliser le RND(TI) classique ou ses formes plus élaborées où l'on ne prend qu'une partie de la valeur de TI. En effet, avec RND(TI) lorsqu'on met en tension le CBM (à condition de l'avoir éteint au moins 10 ou 20 secondes) on obtiendra à chaque fois les mêmes nombres pseudo-aléatoires (par exemple dans une distribution de jeu de cartes, on obtiendra à chaque allumage la même première distribution de cartes, puis la même deuxième distribution, etc. qu'aux précédents allumages du CBM; ce qui bien sûr enlève beaucoup d'intérêt au jeu).

Le "PET/CBM PERSONAL COMPUTER GUIDE" de A. OSBORNE et C.S. DONAHUE (2ème édition, 1980) signale que les séquences générées par RND(0) ne sont pas très bonnes, car les séquences successives se ressemblent un peu (tout en étant différentes). Ce livre conseille, pour avoir des séquences plus aléatoires, de faire :

```
10 X = RND(-RND(0))
```

en tout début de programme pour avoir à chaque utilisation du programme une racine différente. Puis d'utiliser ensuite RND(1) pour générer la séquence pseudo-aléatoire.

Je n'ai trouvé mention de cette différence importante du basic 4.0 dans aucun des textes en français que j'ai pu consulter :

- notices en français fournies par Procep à la livraison de mon 4032 en Décembre 1981.
- livres très récents comme :
  - ."Manuel de l'utilisateur du CBM 8001" (1981) qui pourtant traite spécifiquement du Basic 4.0 mais continue à indiquer la fonction RND des anciens Basics.
  - ."La découverte du PET/CBM" (1981) de D.J.DAVID
  - ."Jeux trucs et comptes" (1981) de M. BENELFOUL
  - ."Clefs pour le Pet/CBM" de D.J.DAVID
  - .etc.

Je n'ai trouvé citée cette nouvelle fonction RND que dans deux manuels en anglais : "User's reference Manual Commodore Basic Version 4.0" de Commodore et "PET/CBM personal computer guide" 2ème édition 1980 de Osborne et Donahue (d'autres livres anglais édités en 1981 n'indiquent encore que la fonction RND des anciens basics).

Dr. Alain TASSY

Le coin du débutant

Opérations sur les fichiers

Les définitions indispensables à tout traitement de fichier.

Un fichier est un recueil d'informations. Ces informations concernent généralement les individus d'un certain ensemble, par exemple les clients d'une entreprise, les informations sont alors le nom, l'adresse, le chiffre d'affaire de l'année en cours,... de chaque client. Pour être utilisable et modifiable un fichier doit avoir une structure nettement définie, "connue" des programmes qui le manipulent directement.

Un fichier va cependant être caractérisé par :

- son support physique.

Les supports informatiques les plus courants sont les cartes perforées, les disques et les bandes magnétiques. Entre les disques de plusieurs centaines de millions de caractères des gros ordinateurs et ceux de quelques centaines de milliers de caractères des micro-ordinateurs il n'y a aucune différence fondamentale : les premiers contiennent plus d'informations, les traitent plus rapidement et de façon plus fiable, mais ils ont exactement les mêmes fonctions. Il en est de même pour les bandes magnétiques d'ordinateurs et les cassettes des micro-ordinateurs.

- son organisation.

L'organisation du fichier va dépendre de son support et va donner une plus ou moins grande souplesse dans la manipulation du fichier.

Les supports séquentiels : ce sont les cartes perforées ou les bandes magnétiques ou cassettes par exemple.

Pour accéder à une information (par exemple l'adresse de Monsieur Schmitt) il est indispensable de défiler le fichier depuis son début jusqu'à l'enregistrement de Monsieur Schmitt.

Les supports à accès sélectif : on dit parfois à tort, à accès direct (traduction de "random access"), cela correspond essentiellement aux disques magnétiques. Sur ces derniers, l'information est placée sur des pistes concentriques et voisines ; elle est écrite ou lue à l'aide d'une tête de lecture-écriture comparable à celle des magnétophones : pour lire ou écrire une information la tête de lecture-écriture est placée en face d'une piste, la lecture ou l'écriture se fait alors comme sur une piste de bande d'un magnétophone. Pour accéder à une information, la tête de lecture-écriture doit d'abord se placer en face de la piste correspondante, d'où le terme d'accès sélectif.

Les organisations de fichier sur les supports à accès sélectifs sont :

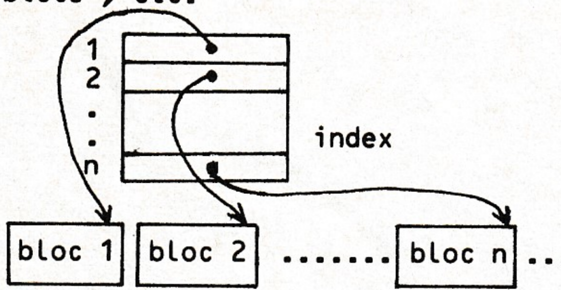
- l'organisation séquentielle : le fichier est sur disque, mais orga-

nisé séquentiellement, il faudra comme sur bande magnétique accéder séquentiellement aux enregistrements successifs.

- l'organisation relative : si le critère de recherche d'une information est sa position relative dans le fichier, pour peu que tous les enregistrements aient la même longueur, le système peut calculer l'adresse de l'enregistrement souhaité sur disque et y accéder directement, c'est à dire sans aucune lecture intermédiaire.

- les organisations dites directes: ce sont les séquentiels indexés, les organisations utilisant des fonctions de "hash-coding", ou des organisations de type "liste".

Le critère de recherche ou de rangement d'un enregistrement est la valeur de l'une de ses rubriques (par exemple le nom du client). Le principe du séquentiel indexé est de découper le fichier en un certain nombre de blocs et d'ajouter un index, table permettant de savoir rapidement dans quel bloc l'enregistrement concerné doit être lu ou écrit. A l'intérieur de chaque bloc l'accès peut être séquentiel ou bien à nouveau indexé avec un découpage du bloc en "sous-blocs", etc.



Les fonctions de "hash-coding" sont destinées à transformer la valeur de la rubrique (on dit clé) d'accès en une adresse d'enregistrement ou de groupe d'enregistrements. Le choix d'une telle fonction peut être délicat et il faut remarquer que l'intérêt d'une telle fonction est entre autres son "pouvoir de réduction", transformant une valeur prise parmi un très grand nombre de valeurs possibles (toutes les combinaisons de lettres sur 20 ou 40 caractères utilisés pour coder un nom par exemple) en une valeur parmi un faible nombre de possibilités (au plus le nombre d'enregistrements réellement prévus pour le fichier). Une telle fonction n'est naturellement pas injective et le problème des doubles (on dit collisions) doit être réglé par des chaînages supplémentaires et parfois par des "zones de débordements" associées au fichier. Dans les organisations de type "liste", une relation d'ordre différente de l'ordre de rangement est établie entre les enregistrements du fichier. Cet ordre est réalisé à l'aide de liens de chaînage entre les enregistrements consécutifs.

Opérations sur les fichiers

Ce qui suit concerne essentiellement les fichiers sur support magnétique.

Les opérations de base sur les fichiers sont :

- la création et l'initialisation du fichier.
- les différentes mises à jour :
  . modification d'une ou plusieurs informations.
  . adjonction de nouveaux enregistrements.
  . suppression de certains enregistrements.
- la suppression du fichier.

La création et l'initialisation d'un fichier.

Dans le cas d'un fichier sur disque, il faut savoir que chaque disque possède un répertoire de tous les fichiers qu'il contient avec un certain nombre d'informations de base sur chacun : nom du fichier, son organisation, sa taille, etc. La création d'un fichier est dans certains cas dissociable de son initialisation et ne représente alors que son insertion dans le répertoire avec la réservation de sa place sur le disque. Les micro-ordinateurs ne peuvent se permettre des procédures aussi aveuglément gourmandes et le fi-

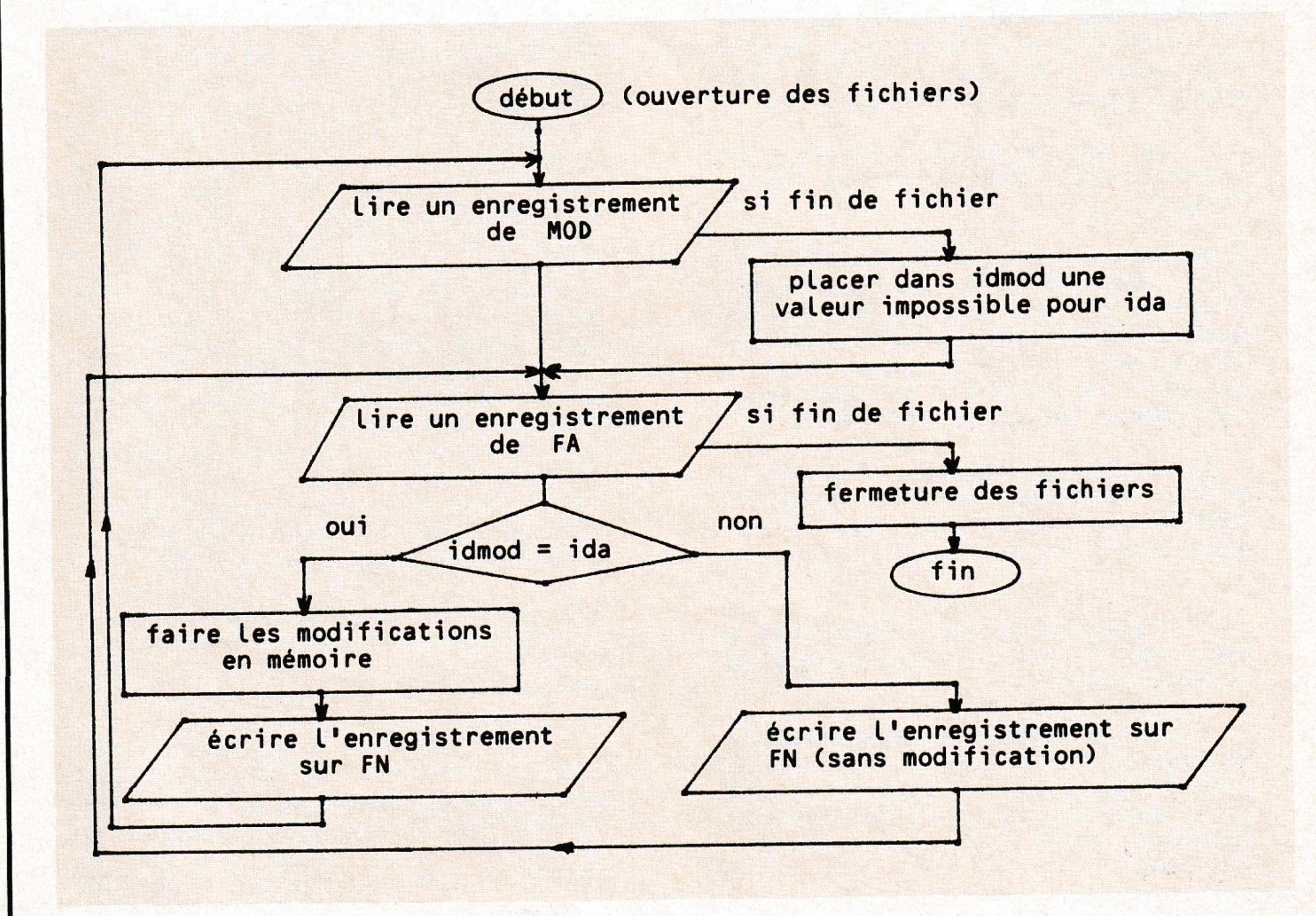
chier n'occupe sur disque que la place qu'il utilise effectivement. L'initialisation est la première écriture du fichier. Compte tenu de son organisation, l'infrastructure du fichier (mise en place des tables d'index dans le cas d'une organisation séquentielle indexée) est fixée lors de l'initialisation.

Les mises à jour

Cas des fichiers séquentiels

Quelque soit la nature de la ou des mises à jour, dès que plusieurs enregistrements sont concernés, il est primordial, au moment de mettre à jour un enregistrement du fichier que tous les enregistrements concernés par la mise à jour et précédant cet enregistrement, aient été effectivement mis à jour. Il n'y a qu'une solution : le fichier à mettre à jour et les ordres de mise à jour doivent être rangés dans le même ordre, sur les valeurs d'une rubrique identifiant sans ambiguité les enregistrements du fichier.

Si la mise à jour ne contient que des modifications des rubriques de certains enregistrements, avec certains systèmes d'entrées/sorties de certains ordinateurs, il est possible de réécrire l'information modifiée à la place de l'ancienne pour peu que l'encombrement soit le même. En général, il vaut mieux recréer un nouveau fichier à partir de l'ancien fichier et des indications de modification. On peut représenter une telle mise à jour par un organigramme. FA est l'ancien fichier, idf est la rubrique identifiante de FA. MOD est le fichier contenant les ordres de modification, idmod est la rubrique identifiante de MOD, c'est la même rubrique (nom de client, par exemple que pour FA. FN est le nouveau fichier.



Sur cet organigramme apparaissent des fonctions "ouverture des fichiers" et "fermeture des fichiers". Le programmeur manipule les fichiers à travers un "système (de gestion) de fichiers" qui se charge des lectures et écritures effectives sur disque. Un de ses buts est de rendre transparente au programmeur une certaine optimisation des temps d'accès et/ou des taux d'encombrement des données sur leur support (cas des bandes magnétiques) : cette optimisation est réalisée en regroupant les enregistrements (logiques) manipulés par le programmeur, en blocs ou enregistrements physiques. Le système de fichiers lit ou écrit sur disques des blocs. Outre une fonction de contrôle non négligeable, l'ouverture du fichier sert à initialiser les algorithmes de fonctionnement du système de fichiers de sorte que les lectures ou écritures (physiques) sur le disque ne soient faites que lorsque cela s'impose. Les lectures écritures du programmeur provoquent essentiellement des transferts en mémoire. La fermeture du fichier sert essentiellement à écrire le dernier bloc d'un fichier utilisé en écriture.

La suppression d'un enregistrement peut être faite de deux façons :

- en marquant l'enregistrement : si l'on peut réécrire sur l'enregistrement on met une valeur remarquable dans une zone réservée. Tôt ou tard, il faut alors réorganiser le fichier en supprimant réellement ces enregistrements.

- en créant un nouveau fichier et en n'y recopiant que les enregistrements qui ne doivent pas être supprimés.

L'algorithme est comparable à celui décrit par l'organigramme de la modification.

L'adjonction d'enregistrements ne peut se faire qu'en recréant un fichier à jour. L'algorithme est alors semblable à celui de la modification, donné sous forme d'organigramme.

Cas des autres organisations de fichiers.

- L'organisation relative peut avoir des mises à jour de type modification ou recevoir de nouveaux enregistrements ajoutés à la suite des plus anciens. L'insertion ou la suppression d'enregistrements n'a à priori pas grand sens.

- Les autres organisations sont en fait des types d'organisations et chaque réalisation de séquentiel indexé, par exemple, adopte des solutions particulières. On peut dire que les suppressions peuvent, selon la réalisation, être faites soit par suppression physique accompagnée de la réorganisation immédiate du fichier, soit par marquage. Dans ce dernier cas la réorganisation peut n'être faite que de temps en temps. Les simples modifications peuvent généralement se faire par réécriture immédiate de l'enregistrement à modifier. Les insertions peuvent être réalisées soit en réorganisant localement le fichier pour insérer le nouvel enregistrement à la place qu'il aurait eue à la création, soit en le plaçant dans une zone de débordement associée au fichier ou au bloc, et en établissant, par chaînage, les liens logiques devant faciliter l'accès au nouvel enregistrement : il s'en suit malgré tout une augmentation des temps d'accès aux enregistrements dans leur ensemble. Lorsque, à la suite d'un certain nombre d'insertions (et/ou de suppressions par marquage), l'augmentation des temps d'accès est exagérée, le fichier doit être réorganisé, afin d'être dans le même état que s'il venait juste d'être créé.

Les utilitaires de traitement de fichiers.

Ce sont des programmes catalogués, souvent donnés ou vendus par le constructeur de l'ordinateur et destinés à faciliter les opérations les plus courantes sur les fichiers. Les utilitaires les plus

courants sont :

- copie, initialisation d'un fichier dans l'organisation choisie, à partir de données situées sur un support quelconque (carte, clavier, bande ou disque magnétique).

- édition sur listing ou sur l'écran des caractéristiques ou du contenu d'un fichier.

- réorganisation d'un fichier

- tri d'un fichier séquentiel

- fusion de deux fichiers séquentiels ordonnés.

Daniel TRECOURT

Les jeux cousus-main

LE JEU DE JOSEPHUS

Le jeu que nous donnons ici est la forme "jeu" du problème de Josephus : N condamnés à mort sont placés en cercle. Etant donné un nombre P inférieur à N, et une position 1 définie au départ, on emmène le P iè, puis le P iè pris à partir de là en tournant dans le même sens autour du cercle restreint, et ainsi de suite ... Seul le dernier restant sera grâcié. Le jeu vous permet de choisir votre place sur le cercle initial. Serez-vous grâcié ?

Le programme affiche le cercle et marque à chaque tour le numéro du tour où chaque condamné est emmené.

Bonne chance !

Jean DELAVILLE

```
70 DD=32768:A=20:B=15:R=10:A=DD+A
80 INPUT"■N";N:INPUT"P";P
90 DIMFG$(N):PH=2*π/N:GOSUB1000
95 INPUT"■VOTRE CHOIX";Q:R=R:I=Q:F1=-7:GOSUB1500:F1=0
100 K=K+1:IFK>NTHENK=1:U=U+1
105 IFFG$(K)="1"THEN100
110 C=C+1:IFC=PTHENC=0:FG$(K)="1":I=K:T=T+1:SK=6+U-7*(U>9):GOSUB1500:SK=0
120 IFT+1<NTHEN100
130 FORI=1TON:IFFG$(I)="1"THEN150
140 PRINT"RESU"I:GOSUB2000:GOTO155
150 NEXT
155 PRINT"■        ■SHIFT■ POUR CONTINUER"
160 WAIT152,1:RUN70
1000 REM/PLACE LES CONDAMNES
1007 I=1:F1=23:GOSUB1500:F1=-11:AD=AD-1:GOSUB1512:F1=0:AD=AD+1
1010 FORI=2TON:GOSUB1500
1020 NEXT:RETURN
1500 W=PH*I:X= A+INT(R*COS(W)):Y=B+INT(R*SIN(W))
1510 AD=X+40*Y
1512 KA=42+SK+F1
1515 POKEAD,42+SK+F1:RETURN
2000 GOSUB1500
2002 POKEAD,42
2005 GETQ$:IFQ$="A"THENRETURN
2007 FORJ=1TO100:NEXT
2010 POKEAD,170
2012 FORJ=1TO100:NEXT
2015 IFQ<>ITHENPRINT"■          PERDU TAPER A POUR LA SUITE":GOTO2000
2020 PRINT"■          GAGNE POUR UNE AUTRE->■A■":GOTO2000
```

Le Tampon Clavier

Une source inépuisable d'astuces

Voici une des sources d'astuces les plus importantes des PET, CBM et VIC. Il y a là, aux adresses mises en jeu près, parfaite compatibilité de toute la gamme Commodore.

Le concept essentiel est que, à mesure que vous tapez sur les touches, le code ASCII de celles-ci est accumulé dans une zone mémoire appelée le tampon-clavier, où elles restent en attente d'être traitées. L'acquisition d'une touche et sa mise dans le tampon clavier est effectuée de façon transparente par rapport à votre programme : elle est effectuée au cours de l'interruption qui se produit tous les soixantièmes de seconde et qui, comme vous le savez, rythme la vie de votre système Commodore.

Quel est l'intérêt ? Eh bien cela permet à votre frappe d'être en avance sur le traitement : si vous tapez sur une touche avant que le programme ne soit en mesure de l'examiner, elle ne sera pas perdue, mais mise en attente dans le tampon. Il y a une limitation : la zone du tampon ne fait que 10 octets, donc vous ne pouvez pas avoir plus de 10 touches d'avance. D'ailleurs, en cas de dépassement, le système a un comportement curieux : lorsque vous tapez sur la onzième touche, tout le tampon est remis à 0 (les dix touches sont perdues) et la onzième touche vient en première position dans le tampon. Mais il y a une probabilité très faible pour que cela arrive : il faut vraiment que vous tapiez vite pour que le système ne vous ait pas rattrapé en dix touches !

Comment est-ce géré ? Très simple. Le tampon forme une zone mémoire de 10 octets d'adresse de départ TC. Il y a d'autre part un "pointeur tampon clavier " qui contient en fait le nombre de touches qui ont été décelées. Appelons TP ce pointeur : le nombre de touches est donc NT = PEEK(TP) et le code ASCII de la dernière touche enregistrée est K=PEEK(TC+NT-1).

Voici les valeurs (en décimal)

	TC	TP
PET 2001	527	525
CBM 3,4,8000	623	158
VIC 20	631	198

Deux applications immédiates :
- pour saisir un caractère sans risquer d'être troublé si l'utilisateur avait appuyé sur une touche antérieurement :

```
100 POKE TP,0
110 GET A$ : IF A$ = "" GOTO 110
```

(C'est intéressant pour les jeux).
- pour attendre qu'on appuie sur une touche quelconque :

```
POKE TP,0 : WAIT TP,1
```

Utilisation élaborée

La clé de l'utilisation astucieuse du tampon clavier est qu'un programme peut - par des POKE dans

le tampon clavier - simuler le fait que l'utilisateur tape sur des touches.

C'est ainsi, par exemple, que procède le système lorsqu'on appuie sur SHIFT et RUN/STOP: à ce moment, il introduit dans le tampon clavier les caractères L O A D Return R U N Return, et le tour est joué: la machine obéit aux commandes directes LOAD puis RUN (N.B.: sur un 4000-8000, c'est DLOAD qui est introduit).

Le point fondamental est que, après vos POKE dans le tampon clavier, il faut sortir du programme par END : le système imprime READY puis regarde ce qu'il y a dans le tampon clavier : s'il y a quelque chose, il le prend en compte.

L'utilisation principale de ceci consiste à ce qu'un programme se modifie lui-même en générant des instructions comme si l'utilisateur venait de les taper. Il y a une difficulté : le fait qu'on est limité à dix caractères. La solution dérive des propriétés de l'édition sur écran :
1) imprimer (par PRINT "...") l'instruction que vous voulez générer.

2) Il faut ensuite faire revenir le curseur sur la ligne que vous venez d'imprimer et faire Return dessus. Cela s'obtient par ... POKE TC,145: POKE TC+1,145 : POKE TC+2,145 : POKE TC+3,13 : POKE TP,4 : END
(145 = ASC(curseur haut)
13 = ASC(Return))
et le tour est joué. On vérifie expérimentalement que c'est trois curseurs haut qu'il faut pour revenir sur la ligne.

Il reste encore une difficulté: puisqu'on a fait END, on n'est plus en exécution du programme. Il faut donc y revenir, ce qui se fait en introduisant dans le tampon clavier la commande directe GOTO numéro. On l'introduira bien sûr sous forme abrégée G shift O.

Attention, dans cette opération toutes les variables du programme sont perdues puisqu'on modifie des instructions. Donc s'il y a des informations à conserver pour quand on reprend l'exécution, il faut les mettre quelque part en mémoire par POKE.

Exemples :
----

1 - Ceci est pleinement mis à profit par le programme LIBRARAN (voir numéro 2 p. 18, numéro 3 p.29) pour les fonctions autonumérotation et suppression. Pour l'autonumérotation, on imprime le numéro généré par le programme, puis la chaîne de caractères qui a été fournie comme corps de l'instruction.

Pour la suppression, on imprime chaque numéro et on fait Return dessus.

Voir spécialement les lignes 16 (sauvegarde de données), 19 (impression de la ligne) et 35 - 37 (remplissage du tampon).

2 - Si l'on veut faire un programme de calcul d'intégrale suffisamment général, il faut pouvoir définir la fonction à intégrer au dernier moment lors de l'exécution. Cela se fait en "générant" sur le moment la ligne DEF FN correspondante :

```
 0 REM PROGRAMME D'INTEGRATION
   GENERAL
 1 INPUT "QUELLE FONCTION VOULEZ-
   VOUS INTEGRER" ; F$
 2 PRINT "clr 10 DEF FNF(X)="F$
 3 TC = 623(CBM)/527(PET)/631(VIC):
   TP=158(CBM)/525(PET)/198(VIC)
 4 POKE TC,ASC("Home") : POKE TC+1,
   13 : POKE TC+2, ASC("G")
 5 POKE TC+3,ASC("o") : POKE TC+4,
   ASC("1") : POKE TC+5,ASC("0")
 6 POKE TC+6,13 : POKE TP,7 : END
10 DEF FNF(X) = EXP(X) : REM SERA
   CHANGEE
20 INPUT "INTERVALLE, NB DE PAS" ;
   A, B, N
30 H = (B-A)/N :
   S=(FNF(A) + FNF(B))/2
40 FOR X=A+H TO B - H STEP H
50 S = S + FNF(X) : NEXT X
60 S = H * S
70 PRINT "INTEGRALE = " ; S
```

A vos claviers pour imaginer d'autres applications.

Si vous en trouvez d'intéressantes ou originales, envoyez les nous.

Daniel-Jean DAVID

Registres d'Entrées - Sorties du VIC

LE BOITIER VIC.6560

HEXA				DEC
9000	ENTRELA-CEMENT	MARGE GAUCHE		36864
9001	MARGE HAUTE			36865
9002	AD.ECRAN BIT 9	NOMBRE DE COLONNES		36866
9003	AD.BALAY BIT 0	NOMBRE DE RANGEES	CARACT. DOUBLES	36867
9004	ADRESSE OU EN EST LE BALAYAGE BIT 8 A BIT 1			36868
9005	ADRESSE MEMOIRE ECRAN BITS 13 A 10	ADRESSE GENE. CAR. BITS 13 A 10		36869
9006	CRAYON LUMINEUX HORIZONTAL			36870
9007	CRAYON LUMINEUX VERTICAL			36871
9008	POTENTIOMETRE X			36872
9009	POTENTIOMETRE Y			36873
900A	VOIX 1			36874
900B	VOIX 2			36875
900C	VOIX 3			36876
900D	VOIX 4 (BRUIT)			36877
900E	MODE MULTI - COULEURS	AMPLITUDE SON		36878
900F	COULEUR FOND ECRAN	CONTRASTE INVERSE	COULEUR CADRE	36879

LE 6522 numéro 1   (interruption NMI)

HEXA									DEC
9110	PB PORT PARALLELE UTILISATEUR ET RS232 (bit 7 : *, bit 3 : +)								37136
9111	PA UTILISE PAR L'ADRESSE 911F								37137
9112	DIRECTION PB								37138
9113	DIRECTION PA								37139
9114	TEMPORISATEUR 1 T1-CL								37140
9115	T1-CH VITESSE EMISSION RS232								37141
9116	T1-LL SYNCHRO. ECRITURE CASSETTE								37142
9117	T1-LH								37143
9118	TEMPORISATEUR 2 T2-L								37144
9119	T2-H SYNCHRO. ENTREE RS232								37145
911A	SR REGISTRE A DECALAGE (INUTILISE)								37146
911B	ACR COM.	T1	COM. T2	C O M M A N D E S R			VERROUILLAGE PB	PA	37147
911C	PCR CONTROLE CB2 EMISSION RS232			CONTROLE CB1	CONTROLE CA2 MOTEUR CASSETTE			CONTROLE CA1	37148
911D	IFR NMI	T1	T2	CB1 ENTREE RS232	CB2	SR	CA1 TOUCHE RESTORE	CA2	37149
911E	IER	T1	T2	CB1	CB2	SR	CA1	CA2	37150
911F	PA SORTIE ATN	TESTE CASSETTE	MANCHES A BALAI : JOY0	JOY1	JOY2	CRAYON LUMINEUX	ENTREES SERIE IEEE DONNEE	HORLOGE	37151

* le bit 7 sert aussi d'entrée JOY 3
+ Le bit 3 sert aussi de sortie écriture cassette.

LE 6522 numéro 2 (interruption IRQ)

<table>
<tr><th>HEXA</th><th colspan="8"></th><th>DEC</th></tr>
<tr><td>9120</td><td colspan="8">PB SELECTION RANGEE CLAVIER</td><td>37152</td></tr>
<tr><td>9121</td><td colspan="8">PA ENTREE COLONNE CLAVIER</td><td>37153</td></tr>
<tr><td>9122</td><td colspan="8">DIRECTION PB</td><td>37154</td></tr>
<tr><td>9123</td><td colspan="8">DIRECTION PA</td><td>37155</td></tr>
<tr><td>9124</td><td colspan="8">T1-CL SYNCHRO. LECTURE CASSETTE ;</td><td>37156</td></tr>
<tr><td>9125</td><td colspan="8">T1-CH RYTHME</td><td>37157</td></tr>
<tr><td>9126</td><td colspan="8">T1-LL INTERRUPTIONS CLAVIER</td><td>37158</td></tr>
<tr><td>9127</td><td colspan="8">T1-LH</td><td>37159</td></tr>
<tr><td>9128</td><td colspan="8">T2-L TIMEOUT BUS IEEE SERIE</td><td>37160</td></tr>
<tr><td>9129</td><td colspan="8">T2-H SYNCHRO. L/E CASSETTE</td><td>37161</td></tr>
<tr><td>912A</td><td colspan="8">SR REGISTRE A DECALAGE (INUTILISE)</td><td>37162</td></tr>
<tr><td>912B</td><td colspan="2">ACR COM. T1</td><td>COM.
T2</td><td colspan="3">C O M M A N D E
S R</td><td colspan="2">VERROUILLAGE
PB | PA</td><td>37163</td></tr>
<tr><td>912C</td><td colspan="3">PCR CONTROLE CB2
SORTIE DONNEE IEEE</td><td>CONTROLE
CB1</td><td colspan="3">CONTROLE CA2
SORTIE HORLOGE IEEE</td><td>CONTROLE
CA1</td><td>37164</td></tr>
<tr><td>912D</td><td>IFR
IRQ</td><td>T1</td><td>T2</td><td>CB1</td><td>CB2</td><td>SR</td><td>CA1</td><td>CA2</td><td>37165</td></tr>
<tr><td>912E</td><td>IER</td><td>T1</td><td>T2</td><td>CB1</td><td>CB2</td><td>SR</td><td>CASSETTE</td><td>CA2</td><td>37166</td></tr>
<tr><td>912F</td><td colspan="8">PA UTILISE PAR L'ADRESSE 9121</td><td>37167</td></tr>
</table>

Daniel-Jean DAVID

Connectez deux magnétos à votre VIC

I - PRINCIPE

Une manip. pas si stupide que cela et qui peut s'avérer très économique.

Voici un article d'une nouvelle forme dans La Commode, mais nous espérons qu'il y en aura beaucoup d'autres de cette manière. Nous appelons cette forme les "articles-dialogue".

Dans un article de cette forme, au lieu de parachuter la solution complète d'un problème dans le style "voilà, nous sommes les plus forts, il faut faire ceci et cela...", nous donnons d'abord le principe d'une manip. Nous laissons ensuite le temps à nos lecteurs d'expérimenter sur la manip. et de nous livrer leurs réflexions, les précautions dont ils auront découvert la nécessité, les utilisations qu'ils en auront faites.

Nous pensons que c'est ainsi, à propos d'une manipulation précise, qu'un véritable dialogue pourra s'établir entre La Commode et les lecteurs et que La Commode pourra au mieux remplir sa mission.

Mais entrons dans le vif du sujet...

Pourquoi deux magnétos ?

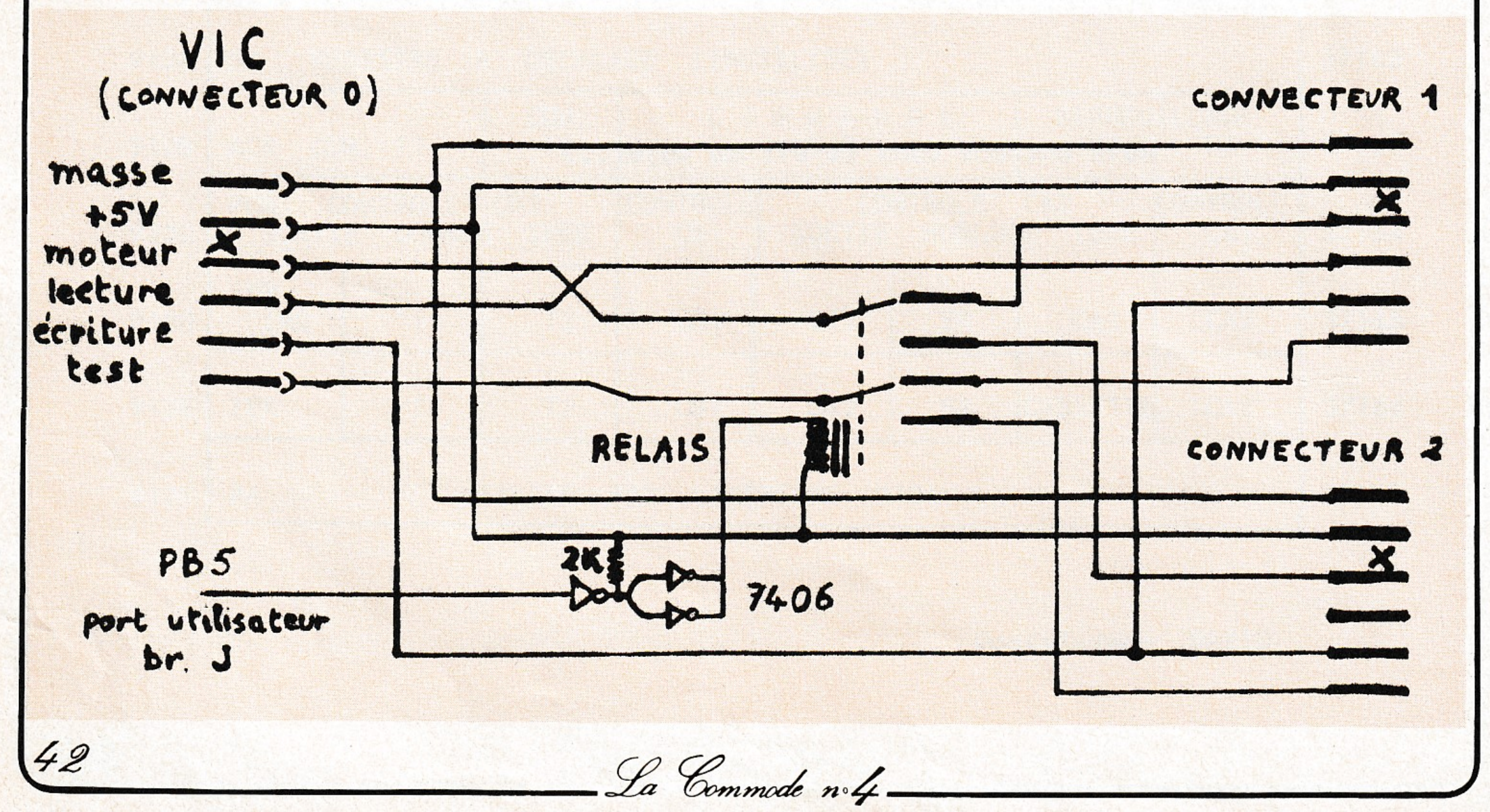
Toute utilisation un peu sérieuse de fichiers de données sur cassettes nécessite deux magnétos. Nous ne discutons pas ici de la possibilité de fichiers sérieux sur cassette : il est clair que les disquettes sont préférables, mais leur prix (même celui de la mono-disquette VIC 1540) peut s'avérer trop élevé pour certains. La manip. que nous proposons ne nécessite que l'achat d'un second magnéto à 500 Frs. environ. Pourquoi un second magnéto ? Parce que tout fichier de données a un jour ou l'autre besoin d'être mis à jour : des données doivent être modifiées, ajoutées, supprimées. Or, ceci est impossible sur cassette : si vous modifiez sur place un enregistrement, des données situées en avant vont se trouver effacées. La solution ? Disposer d'un second magnéto et procéder par recopie : le premier magnéto est utilisé en lecture et il porte l'ancien fichier ; le second porte le nouveau fichier en écriture. Lorsqu'un enregistrement est sans modification, il est lu sur l'ancien, et réécrit sur le nouveau fichier. Lorsqu'il y a une modification, il est lu sur l'ancien fichier pour être passé; et on écrit la nouvelle version sur le nouveau fichier.

Ceci est assez facile à mettre en oeuvre sur PET ou CBM puisque ces machines ont deux connecteurs magnéto. C'est plus difficile sur le VIC qui ne prévoit qu'un magnéto possible. Le but de cet article est précisément d'essayer de surmonter ce problème. On va multiplexer le connecteur sur deux connecteurs identiques. Le connecteur original du VIC sera appelé 0, les deux connecteurs où se brancheront réellement les magnétos seront appelés 1 et 2.

Hardware

Pour simplifier, puisqu'on a besoin d'un magnéto en lecture et un en écriture, on supposera que 1 sera toujours en lecture et 2 toujours en écriture. Donc, on pourra faire les liaisons permanentes 0.lecture - 1.lecture et 0.écriture - 1.écriture. Mais il est facile, si l'on veut, de s'affranchir de cette restriction. On a en outre les liaisons évidentes 0.masse - 1.masse et 2.masse et 0.+5V - 1.+5V et 2.+5V.

Les seules lignes à multiplexer sont donc test (teste si une touche du magnéto est enfoncée) et moteur. Pour cela, nous proposons un relais double inverseur commandé par une broche du port utilisateur (par l'intermédiaire d'un 7406 ou 7416 qui joue le rôle d'ampli). Le schéma est donc le suivant :



Les deux lignes d'écriture ont été reliées (c'est comme cela dans les CBM) pour permettre d'utiliser le connecteur 1 comme connecteur normal sans avoir à démonter le multiplexeur. Il faut alors assurer que ce sont les lignes du connecteur 1 qui travaillent lorsque le relais est au repos. Nous utilisons un relais 5-6 V, impédance de bobine 100 ohms (en fait un ITT 240 AWO).

Software

Nous utilisons pour la commande, la ligne PB5 du port utilisateur, car elle n'est pas utilisée par le RS232. Cette ligne doit être mise en sortie par

POKE 37138, PEEK (37138) OR 32

Le magnéto 1 est actif si le bit 5 est à 1 d'où

POKE 37136, PEEK (37136) OR 32

Le magnéto 2 est activé, en mettant le bit à 0 d'où

POKE 37136, PEEK (37136) AND 223

Les opérations sur l'un ou l'autre magnéto doivent se faire en utilisant le même numéro de périphérique égal à 1. A la différence du PET/CBM, 2 n'est pas un magnéto (c'est le RS 232).Donc après avoir activé le magnéto voulu, on lit sur le magnéto 1 par :

- OPEN 1,1,0,"nom"
- INPUT≠1,X
- CLOSE 1

et on écrit sur 2 par :

- OPEN 2,1,1,"nom"
- PRINT≠2,X
- CLOSE 2

Mais ce n'est pas tout. Vous savez que le magnéto fonctionne grâce à une zone tampon en mémoire de 192 octets. (d'adresse 828 à 1019 - 166 contient le pointeur de gestion de ce tampon = nombre d'octets où on en est).

Eh bien, il faut multiplexer ce tampon, sinon les données lues se mélangeront aux données à écrire. Il faut donc une zone de sauvegarde du tampon pour le magnéto numéro 1 et une zone pour le magnéto 2. Nous vous suggérons les adresses suivantes :

. 7168 à 7359 ($1C00 - 1CBF) = tampon numéro 1

. 7360 à 7551 ($1CC0 - 1D7F) = tampon numéro 2

. 7552 ($1D80) = pointeur numéro 1

. 7553 ($1D81) = pointeur numéro 2

Il faut, bien sûr, diminuer la taille mémoire disponible par POKE 56,28 (pour un VIC de base).

Avant chaque opération sur un magnéto, il faut sauver le tampon de l'autre magnéto et charger la zone tampon avec la sauvegarde du magnéto à activer.

Nous suggérons les routines suivantes :

Initialisation

```
1000 POKE 56,28 : CLR
1010 POKE 37138, PEEK (37138) OR 32
1020 POKE 7552,0 : POKE 7553,0 :
     RETURN
```

Activation magnéto 1

```
1100 POKE 37136, PEEK(37136) OR 32
1110 POKE 7553, PEEK(166) :
     POKE166, PEEK(7552)
1120 A = 828 : B = 7360 :
     GOSUB 1300
1130 A = 7168 : B = 828 :
     GOSUB 1300
1140 RETURN
```

Activation magnéto 2

```
1200 POKE 37136, PEEK(37136) AND
     223
1210 POKE 7552, PEEK(166) :
     POKE 166, PEEK (7553)
```

```
1220 A = 828 : B = 7168 :
     GOSUB 1300
1230 A = 7360 : B = 828 :
     GOSUB 1300
1240 RETURN
```

Transfert tampon

```
1300 FOR I = 0 TO 191
1310 POKE B + I, PEEK (A+I)
1320 NEXT : RETURN
```

Conclusion

Essayez ces routines. Envoyez-nous vos remarques. Il y a probablement des précautions que nous avons oubliées ou des astuces à trouver.

A SUIVRE...

Daniel-Jean DAVID



LES ADRESSES STRATEGIQUES DU VIC

HEXA	DEC	DESCRIPTION
0000-0002	0-2	Saut pour USR (adresse en 1-2)
0003-0004	3-4	vecteur pour conversion réel-entier
0005-0006	5-6	vecteur pour conversion entier-réel
0007	7	caractère recherché
0008	8	indicateur mode guillemets
0009	9	sauvegarde colonne de tabulation
000	10	0=LOAD 1=VERIFY
000B	11	pointeur dans tampon d'entrée/numéro d'indice
000C	12	indicateur de DIM par défaut
000D	13	type : $FF = chaîne 00 = numérique
000E	14	type : $80 = entier 00 = réel
000F	15	indicateur mémoire, mode guillemets en LIST, balayage DATA
0010	16	indicateur indice / FNx
0011	17	0=INPUT, $40 = GET, $98 = READ
0012	18	signe pour ATN/indicateur de comparaisons
0013	19	indicateur actuel d'interrogation d'entrée
0014-0015	20-21	valeur entière
0016	22	rangement temporaire du pointeur de pile
0017-0018	23-24	dernier vecteur de chaîne temporaire
0019-0021	25-33	pile pour chaînes temporaires
0022-0025	34-37	zone de pointeurs utilitaires
0026-002A	38-42	zone pour le produit dans une multiplication
002B-002C	43-44	pointeur : début de BASIC
002D-002E	45-46	pointeur : début des variables
002F-0030	47-48	pointeur : début des tableaux
0031-0032	49-50	pointeur : fin des tableaux
0033-0034	51-52	pointeur : zone des chaînes
0035-0036	53-54	pointeur utilitaire chaînes
0037-0038	55-56	pointeur : fin de mémoire
0039-003A	57-58	numéro de ligne BASIC courante
003B-003C	59-60	numéro de ligne BASIC précédente
003D-003E	61-62	pointeur instruction BASIC pour CONT
003F-0040	63-64	numéro de ligne DATA courante
0041-0042	65-66	adresse courante de DATA
0043-0044	67-68	vecteur pour INPUT
0045-0046	69-70	nom de la variable courante
0047-0048	71-72	adresse de la variable courante
0049-004A	73-74	pointeur vers variable pour FOR...NEXT
004B-004C	75-76	sauvegarde de Y ou d'opérateur ou du pointeur BASIC

HEXA	DEC	DESCRIPTION
004D	77	accumulateur des symboles de comparaison
004E-0053	78-83	zone de travail diverse
0054-0056	84-86	vecteur de saut pour fonctions
0057-0060	87-96	zone de travail numérique diverse
0061	97	Acc. numéro 1 : exposant
0062-0065	98-101	Acc. numéro 1 : mantisse
0066	102	Acc. numéro 1 : signe
0067	103	pointeur vers constantes pour évaluer une série
0068	104	débordement de Acc. numéro 1
0069-006E	105-110	Acc. numéro 2 : exposant, mantisse, signe
006F	111	signe de la comparaison Acc.1 Acc.2
0070	112	arrondi de l'Acc. numéro 1
0071-0072	113-114	longueur tampon-cassette / pointeur série
0073-008A	115-138	sous-programme CHRGET (obtient un caractère Basic)
007A-007B	122-123	pointeur Basic (dans CHRGET)
008B-008F	139-143	racine pour RND
0090-	144	mot d'état ST
0091	145	copie du PIA clavier : indicateurs de STOP et RVS
0092	146	constante de vitesse pour cassette
0093	147	0 = LOAD ; 1 = VERIFY
0094	148	sortie série : indicateur de caractère différé
0095	149	caractère série différé
0096	150	EOT reçue sur bande
0097	151	sauvegarde de registre
0098	152	nombre de fichiers ouverts
0099	153	périphérique standard d'entrée (normalement 0)
009A	154	périphérique de sortie standard déterminé par CMD (normalement 3)
009B	155	parité du caractère sur cassette
009C	156	indicateur de caractère reçu
009D	157	contrôle de sortie : directe = $80, RUN = 0
009E	158	témoin d'erreur sur cassette passe 1/tampon caractère
009F	159	témoin d'erreur corrigée sur cassette (passe 2)
00A0-00A2	160-162	horloge au 60e de seconde
00A3	163	compte de bits série / indicateur d'EOI
00A4	164	compteur de cycles
00A5	165	décompteur écriture cassette / compteur de bits
00A6	166	pointeur dans tampon cassette
00A7	167	compteur en-tête écriture cassette / passe lecture/ bit entrée
00A8	168	nouvel octet écriture cassette / erreur lecture / compte bits entrées
00A9	169	écrit bit départ/ erreur lecture bit / bit de départ
00AA	170	balayage cassette / assemblage octet
00AB	171	longueur en-tête écriture / checksum lecture / parité
00AC	172-173	pointeur : tampon cassette, déroulement écran
00AE-00AF	174-175	adresse fin cassette / fin du programme
00B0-00B1	176-177	constantes de vitesse pour cassette
00B2-00B3	178-179	pointeur : début du tampon cassette
00B4	180	1 = temporisateur cassette activé /compte de bits
00B5	181	EOT cassette / prochain bit à envoyer en RS232
00B6	182	erreur caractère en lecture / tampon octet à sortir
00B7	183	nombre de caractères du nom de fichier
00B8	184	fichier logique courant
00B9	185	adresse secondaire courante
00BA	186	périphérique courant
00BB-00BC	187-188	pointeur vers nom de fichier
00BD	189	mot à décaler en sortie / caractère entré
00BE	190	nombre de blocs restant à lire ou écrire
00BF	191	tampon mot série

HEXA	DEC	DESCRIPTION
00C0	192	commande moteur cassette
00C1-00C2	193-194	adresse départ d'E/S
00C3-00C4	195-196	pointeur routine de préparation du noyau
00C5	197	dernière touche appuyée
00C6	198	nombre de caractères dans le tampon clavier
00C7	199	indicateur de contraste inversé
00C8	200	pointeur vers fin de ligne en entrée
00C9-00CA	201-202	témoin curseur en entrée (ligne/colonne)
00CB	203	touche enfoncée : 64 si aucune
00CC	204	0 = le curseur clignote
00CD	205	décompteur pour vitesse de clignotement curseur
00CE	206	caractère sous le curseur
00CF	207	phase de clignotement curseur
00D0	208	entrée depuis l'écran ou le clavier
00D1-00D2	209-210	pointeur vers ligne écran
00D3	211	position du curseur dans la ligne
00D4	212	0 = curseur direct ; sinon programmé
00D5	213	longueur ligne courante sur écran
00D6	214	ligne où se trouve le curseur
00D7	215	tampon / checksum / dernière touche
00D8	216	nombre d'insertions en attente
00D9-00F0	217-240	table des liens des lignes-écran
00F1	241	lien bidon
00F2	242	marqueur ligne écran
00F3-00F4	243-244	pointeur mémoire couleur écran
00F5-00F6	245-246	pointeur clavier
00F7-00F8	247-248	pointeur vers réception RS232
00F9-00FA	249-250	pointeur vers émission RS232
0100-01FF	256-511	pile processeur, dont sont utilisés temporairement :
00FF-010A	255-266	zone de travail pour la conversion réel ASCII
0100-013E	256-318	mémorisation erreurs cassette
0200-0258	512-600	tampon d'entrée du Basic
0259-0262	601-610	table des fichiers logiques
0263-026C	611-620	table des numéros de périphériques
026D-0276	621-630	table des adresses secondaires
0277-0280	631-640	tampon clavier
0281-0282	641-642	plus petite adresse de RAM Basic
0283-0284	643-644	plus grande adresse de RAM Basic
0285	645	indicateur de time-out sur le bus série
0286	646	code couleur courant
0287	647	couleur sous le curseur
0288	648	page de la mémoire d'écran
0289	649	taille maximum du tampon d'écran (doit être X 10)
028A	650	répétition automatique des touches : 0 = touches curseur ; 127 = aucune touche 128=toutes les touches
028B	651	compteur pour la vitesse de répétition
028C	652	compteur du délai pour la répétition
028D	653	indicateur de SHIFT, C=, CTRL : 0=rien, 1=SHIFT, 2=C=, 4= CTRL
028E	654	dernier motif de SHIFT
028F-0290	655-656	pointeur vers la routine d'établissement de la table de transcodage clavier
0291	657	mode des touches (Katakana ou non ; aucune utilité dans les versions européennes)
0292	658	0 = déroulement écran autorisé
0293	659	contrôle du boîtier VIC
0294	660	commande du boîtier VIC
0295-0296	661-662	temps d'un bit pour le RS232
0297	663	état du RS232
0298	664	nombre de bits à envoyer

HEXA	DEC	DESCRIPTION
0299-029A	665-666	code vitesse du RS232
029B	667	pointeur réception RS232
029C	668	pointeur entrée RS232
029D	669	pointeur émission RS232
029E	670	pointeur sortie RS232
029F-02A0	671-672	sauvegarde IRQ durant E/S cassette
0300-0301	768-769	pointeur message d'erreur
0302-0303	770-771	pointeur vers reset tiède de Basic
0304-0305	772-773	pointeur vers routine de codage des mots-clés Basic
0306-0307	774-775	pointeur vers routine d'impression des codes des mots-clés
0308-0309	776-777	pointeur vers départ nouvelle ligne Basic
030A-030B	778-779	pointeur vers routine d'obtention d'un élément arithmétique
030C-030F	780-783	zone à partir de laquelle sont chargés A, X, Y et P lors d'un SYS.
0314-0315	788-789	vecteur interruption hardware ($EABF)
0316-0317	790-791	vecteur de BRK (FED2)
0318-0319	792-793	vecteur de NMI (FEAD)
031A-031B	794-795	vecteur pour OPEN (F40A)
031C-031D	796-797	vecteur pour CLOSE (F34A)
031E-031F	798-799	vecteur pour établir l'entrée standard (F2C7)
0320-0321	800-801	vecteur pour établir la sortie standard (F309)
0322-0323	802-803	vecteur pour restaurer les E/S standard (F3F3)
0324-0325	804-805	vecteur pour INPUT (F20E)
0326-0327	806-807	vecteur pour sortie (F27A)
0328-0329	808-809	vecteur pour tester STOP (F770)
032A-032B	810-811	vecteur pour GET (F1F5)
032C-032D	812-813	vecteur pour abonner les E/S (F3EF)
032E-032F	814-815	vecteur pour USR (FED2)
0330-0331	816-817	pointeur pour LOAD
0332-0333	818-819	pointeur pour SAVE
033C-03FB	828-1019	tampon cassette
0400-0FFF	1024-4095	zone d'extension RAM 3K
1000-1FFF	40096-8191	mémoire RAM d'un VIC de base
2000-7FFF	8192-32767	zone d'extension mémoire

Les quelques "trous" qui restent, seront comblés dans un prochain numéro.

Daniel-Jean DAVID

VIC A BRAC

Des petites choses ... mais bien utiles.

CHR$(14)-CHR$(142)

Sur le VIC, de la même façon que sur CBM8000 et CBM4000-G:
PRINT CHR$(14) fait passer en mode texte (minuscules),
Print CHR$(142) fait passer en mode graphique.

Rappelons que c'est le même effet que la combinaison SHIFT C=. Notons aussi que cette combinaison n'est pas incorporable dans une chaîne de caractères pour être exécutée en mode différé. Il faut donc utiliser le CHR$.

Touches STOP/RESTORE

La combinaison STOP-RESTORE est très importante: elle effectue un RESET restreint de l'ordinateur: il y a vidage de l'écran, retour aux couleurs standard du bord et du fond de l'écran, retour au silence des générateur sonores.

Cette dernière fonction est essentielle vu le mode de fonctionnement des générateurs de sons: ceux-ci sont déclenchés par le POKE convenable et ils continuent ensuite à fournir leur son jusqu'à un nouveau POKE. Si donc vous avez arrêté un programme par un simple STOP alors qu'il avait déclenché un son, le son continuera indéfiniment. La combinaison STOP-RESTORE rétablira un silence salutaire.

Un avantage essentiel de la combinaison STOP-RESTORE est qu'elle ne détruit pas le contenu de la mémoire: en particulier votre programme BASIC est conservé.

La combinaison STOP-RESTORE est très utile pour sortir d'une instruction INPUT (voir rubrique suivante).

Enfin nous avons gardé pour la fin un des effets les plus fondamentaux de la combinaison STOP-RESTORE: elle est capable de faire sortir d'un plantage d'une routine en langage machine, alors que sur le PET/CBM il n'y a pas d'issue. Essayez:

```
10 POKE 5120,76:POKE 5121,0:POKE 5122,20
20 SYS 5120
```

Lorsque vous tapez RUN, le curseur disparait: vous êtes planté (bien sûr: on n'a rien fait d'autre qu'un JMP $1400 en 1400(5120): comme boucle, on ne fait pas mieux!).

La combinaison STOP-RESTORE vous redonne le contrôle et vous pouvez vérifier par LIST que le programme est conservé. Les variables sont d'ailleurs conservées aussi.

INPUT

Sur le VIC, le défaut bien connu que l'instruction INPUT présentait sur PET/CBM a disparu: si à INPUT A vous répondez par RETURN tout court, le VIC considère que A garde sa valeur d'avant INPUT et il continue le programme (alors que sur PET/CBM, le programme était interrompu).

En revanche, comme sur PET/CBM la touche STOP est inopérante en cours d'INPUT. Alors, comment sortir d'un INPUT? Très simple: utilisez la combinaison STOP-RESTORE. Mais alors vous ne pouvez plus reprendre par CONT (vous pouvez faire GOTO numéro del'INPUT).

Pierre-Etienne THALBERG

TRANSLATE OR NOT TRANSLATE

Comme vous savez, le VIC a l'adresse de début du BASIC qui varie avec la configuration. Pour qu'un programme s'adapte automatiquement à toute configuration, ou pour qu'un programe de PET/CBM soit adaptable, la routine de chargement à partir de cassette ou disquette du VIC possède la propriété d'éffectuer une translation d'office du programme afin qu'il commence au début du Basic.

Mais cette translation automatique est nuisible dans certains cas, notamment pour charger une routine en langage machine qui doit toujours rester à la même adresse.

Eh bien, il y a moyen d'inhiber cette translation automatique ; il suffit de faire

LOAD"NOM",périph,mode avec périph = 1(cassette) ou 8(disque)
et mode= 0(translation) ou 1(pas de translation).

Les valeurs par défaut sont périph = 1 et mode = 0.
Ainsi LOAD "ZOZO" charge le programme ZOZO depuis la cassette en effectuant la translation éventuelle voulue.

LOAD "ZOZO",1,1 n'effectue pas de translation.
LOAD "0 : ZOZO",8,1 charge ZOZO à partir de la disquette VIC 1540 et n'effectue pas de translation.

Daniel-Jean DAVID

ENCORE UN PROBLEME AVEC INPUT

Il semble qu'avec les ROM actuellement livrées, l'instruction INPUT"texte";variables pose un problème lorsque le texte dépasse une ligne : le VIC incorpore dans la réponse la partie du texte qui se trouve sur la ligne, et, bien sûr, dit REDO FROM START comme si c'était du numérique qu'on voulait.

Deux solutions possibles :

1) Terminer le texte par un curseur bas (qui sera listé comme un Q inversé.)

2) Faire PRINT "texte" puis INPUT.

Ces solutions s'appliquent à un certain nombre de programmes de "La découverte du VIC".

Un truc qui nous est souvent demandé

Pour positionner le curseur ligne L, colonne K :

```
1000 POKE 781, L-1 : POKE 782, K-1: POKE 783,0
1010 SYS 16*4095 : RETURN
```

Exemple : pour imprimer un X en ligne 10, colonne 20 :

```
L=10 : K=20 : GOSUB 1000 : PRINT "X".
```



Le VIC forme maintenant un système complet.

Connexion Péritel

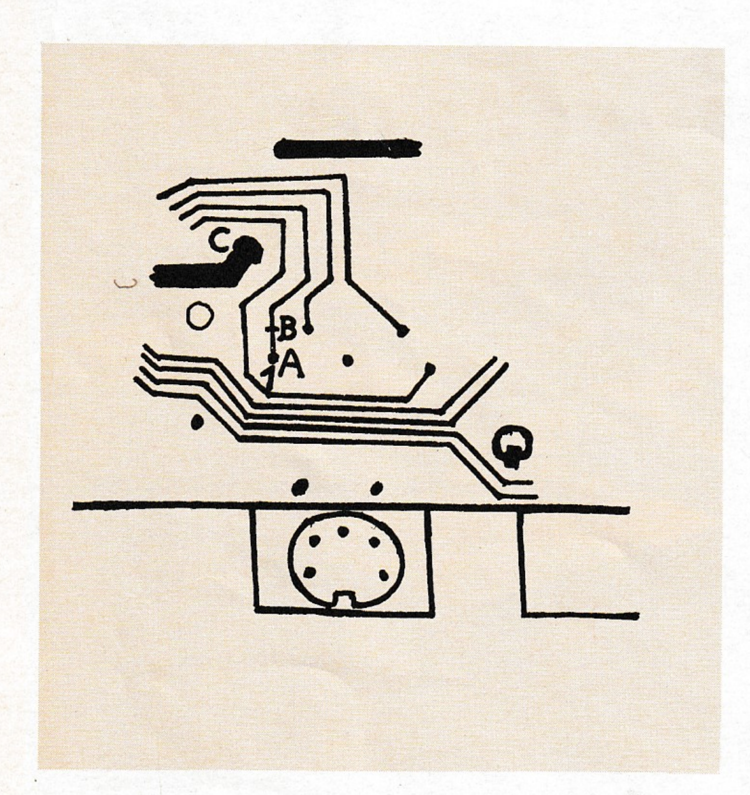
La meilleure solution pour connecter votre VIC à un téléviseur (PAL pour le moment, mais le SECAM arrive) est d'utiliser la prise Péritel (tous les téléviseurs vendus depuis un an l'ont obligatoirement). Les couleurs sont bien meilleures, le son est parfait et il n'y a aucun réglage à effectuer.

Le premier branchement à essayer est le suivant :

branchement 1

VIC connecteur vidéo	TELE prise Péritel
1 +5V	8 (branchement 2 seulement)
2 masse	18
3 son	6
4 vidéo basse	
5 vidéo haute	20

Pour certains téléviseurs, notamment japonais, le branchement 1 ne fonctionne pas. Le problème est qu'il faut faire savoir au téléviseur qu'on utilise la prise Péritel. Pour cela, il faut alimenter la broche Péritel-8. Le branchement 2 consiste à ajouter la liaison VIC vidéo 1 - Péritel 8. La broche VIC vidéo 1 étant alimentée en +5V.



Il y a des cas où le branchement 2 ne fonctionne pas. C'est parce que, pour certains téléviseurs, le +5V est insuffisant. La solution va être de remplacer sur la broche VIC-vidéo 1, le +5V par du 9V non régulé qui existe dans le VIC.

Il faut pour cela que vous (mais votre distributeur acceptera volontiers de s'en charger) ouvriez votre VIC, et dégagiez la plaque circuit imprimé. La figure ci-contre représente la zone de cette plaque autour du connecteur vidéo, vue de dessous (côté soudures) :

Il faut couper la trace en B et relier A et C. Faites exactement ce qui est dit, car il ne faut pas envoyer du 9 V n'importe où.

Pierre-Etienne THALBERG

Affichage haute résolution

INTRODUCTION GENERALE

Nous avons pris une bonne résolution : nous vous disons comment faire de la haute résolution.

Enfin, nous commençons cette série promise depuis longtemps. La série sera en fait double, se subdivisant entre les utilisateurs de VIC et les autres.

Pour les utilisateurs de VIC, on indiquera comment utiliser les possibilités de haute résolution déjà présentes dans le hardware du VIC.

Pour les autres, nous décrirons la réalisation d'une carte haute résolution que nous avons faite, à ajouter dans un CBM 3000. Pour ceux qui ne sont pas très bricoleurs, rappelons que dans notre rubrique bancs d'essai, vous trouverez la description et l'appréciation de différentes cartes haute résolution en vente sur le marché.

Le premier article de cette série est commun à toutes les machines : il décrit comment fonctionne l'affichage d'un micro-ordinateur.

Que ce soit sur un téléviseur ou sur un moniteur incorporé à l'ordinateur, le principe fondamental est que l'image est décomposée en une matrice de points élémentaires, chaque point étant susceptible d'être allumé ou éteint. Nous faisons ici abstraction de la question de la couleur. Les points forment un réseau de P lignes élémentaires sur Q colonnes élémentaires ; il y a donc N = P x Q points avec les valeurs :

	VIC-20	PET/CBM 2000 3000/4000	CBM 8000
P	184	200	200
Q	176	320	640
N	32384	64000	128000
N/8	4 K	8 K	16 K

Le téléviseur ou le moniteur balaie l'image, ligne élémentaire par ligne élémentaire. Pour chaque point de l'image, au moment où il le balaie, il doit être capable de savoir s'il est allumé ou éteint. Le fait d'être allumé ou éteint pouvant être codé sur un bit, (1 = allumé, 0 = éteint), ceci représente une information, donc une mémoire de N = P x Q bits, ou, si l'on compte en octets N/8 octets. On voit sur le tableau ci-dessus que cela représente approximativement 4K octets pour le VIC-20, 8K octets pour le PET/CBM et 16K octets pour le CBM 8000, ce qui est une taille considérable.

Parlons maintenant des problèmes de vitesse et d'abord de la bande passante que doit admettre le téléviseur. Avec le VIC, on a 25 images par seconde. On doit donc pouvoir passer d'un point allumé à un point consécutif éteint 25 x N fois par seconde soit, comme une période est formée d'une alternance complète on a 25 x N/2 périodes par seconde d'où une fréquence de 16 x 25 = 400 kHz. Sur le PET/CBM 2000 à 4000, le même calcul donne (comme

il y a 60 images/s) 32x30 = 960kHz et sur le CBM 8000, on doit passer 1,9 MHz.

Au point de vue temps d'accès de la mémoire d'écran, comme on accède à 8 points élémentaires à la fois, on doit avoir un accès mémoire à la fréquence de :
VIC : 100 kHz donc toutes les 10 µs
PET/CBM : 240 kHz donc toutes les 4 µs
CBM 8000 : 480 kHz donc toutes les 2 µs
ce qui est compatible avec les mémoires actuelles.

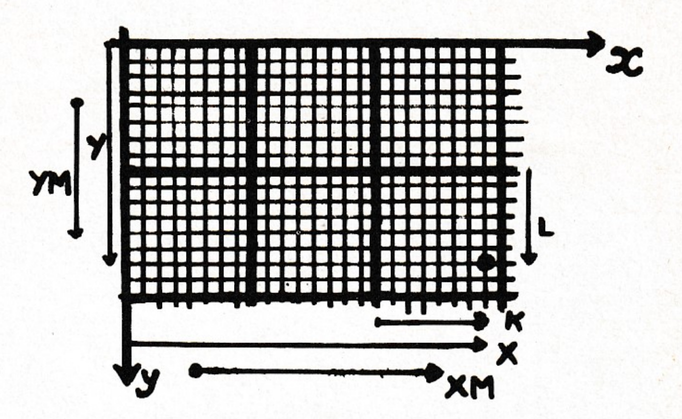
Le problème pour les affichages courants est que les tailles de mémoire vive exigées sont trop importantes (sans compter, en outre, le temps que cela prendrait de remplir cette mémoire pour obtenir un affichage): 4K otés des 5,5 K d'un VIC, c'est trop.

Lorsque l'on ne veut afficher que des caractères sur l'écran (affichage alphanumérique), il y a un moyen de diminuer fortement les besoins en mémoire vive. En revanche, lorsqu'il s'agit d'obtenir un affichage graphique, la simplification n'est pas possible et il faut en revenir aux besoins en mémoire qu'on a vus. Nous allons maintenant expliquer le processus de simplification utilisé dans les affichages alphanumériques, donc dans les VIC et PET/CBM. En somme, pour obtenir un affichage haute résolution, il va falloir revenir en arrière par rapport au processus d'économie de mémoire qui a été employé. Il est donc essentiel de bien comprendre ce processus.

Pour économiser la mémoire, on regroupe les points élémentaires en rectangles de 8 lignes élémentaires sur 8 colonnes élémentaires appelés les "mailles". Un caractère vient s'afficher sur l'écran dans une maille. Les mailles sont réparties en lignes et colonnes. Il faut bien distinguer ces lignes et colonnes qu'on appellera "macroscopiques" des lignes et colonnes élémentaires. Il y a NL lignes et NC colonnes macroscopiques, avec :

	VIC	2000-4000	CBM 8000
NL	23	25	25
NC	22	40	80

Etant donné un point de coordonnées élémentaires X et Y (cf. fig. ci-dessous, origine de numérotation 0), ses macro-coordonnées sont : XM = INT(X/8) et YM = INT(Y/8). Ainsi, le point grossi sur la figure a pour coordonnées élémentaires X = 23 , Y = 14. Il appartient à la maille de coordonnées XM = 2 ; YM = 1. Par rapport à la maille, il est sur la ligne élémentaire L = 6 (reste de la division de Y par 8) et la colonne élémentaire K = 2 (reste de la division de X par 8).



Maintenant, on fait la remarque que les caractères que l'on peut afficher dans une maille appartiennent à un jeu restreint : alors qu'il y a 2↑64 combinaisons possibles dans une maille (il y a 64 points, chacun pouvant être allumé ou éteint), on n'affiche en fait que 256 caractères possibles (y compris les graphiques et les caractères inversés). On va alors coder l'information en deux fois.

Dans une première mémoire (vive, obligatoirement), de nombre d'octets égal au nombre de mailles sur l'écran (donc 506 pour le VIC, 1 K pour le PET/CBM, 2 K pour le CBM 8000), on mettra le code du caractère qui doit figurer dans chaque maille. Si E est l'adresse d'origine de cette mémoire (VIC de base 7680, PET/CBM 32768), l'adresse correspondant à la maille de coordonnées XM,YM est: M=E+NC*YM+XM.

Dans l'exemple ci-dessus, on aurait : M = 7680 + 24 = 7704 (VIC de base) ou M = 32768 + 42 = 32810 (PET/CBM) ou M = 32768 + 82 = 32850 (8000). Si dans cette maille, on veut afficher la lettre A, on y mettra le code-écran de A, donc 1.

On a ensuite une seconde mé-

moire (qui peut être de la mémoire morte) qui va contenir la correspondance code-dessin. Cette mémoire s'appelle le générateur de caractères. Appelons G son origine (32768 sur le VIC à la mise sous tension / indéfinie dans les PET-CBM puisque, dans ces systèmes, elle n'est pas dans l'espace adressable).

Il y a dans cette mémoire 8 octets pour chaque caractère. A l'adresse G + 8 * R + L se trouve un octet qui est l'image de la ligne élémentaire L qu'il faut pour dessiner le caractère de code écran R.

Par exemple, pour la lettre A, on a :

adresse	motif binaire	soit
G+8*1+0	00011000	...●●...
" +1	00100100	..●..●..
" +2	01000010	.●....●.
" +3	01111110	.●●●●●●.
" +4	01000010	.●....●.
" +5	01000010	.●....●.
" +6	01000010	.●....●.
" +7	00000000	....

On voit comment cela fait dessiner un A (N.B. les motifs binaires de l'exemple proviennent du VIC).

En somme, à un instant donné, le système "sait" qu'il a à visualiser le point élémentaire X,Y. Il calcule XM et YM ainsi que K et L. Il obtient donc M, adresse dans la mémoire d'écran. Une lecture dans la mémoire d'écran fournit R, code du caractère à afficher dans la maille. R combiné avec L donne l'adresse où lire, dans le générateur de caractères, le motif binaire qui définit la ligne élémentaire à afficher.

Dans le VIC, ces opérations avec notamment les deux lectures en mémoire, sont orchestrées par le boîtier VIC 6561. Dans les CBM 4000 et 8000 elles sont effectuées par le contrôleur d'écran 6845/6545. Dans le PET ou le CBM 3000, elles sont effectuées par hardware : on ne lit qu'une fois le motif binaire d'une ligne élémentaire et un registre à décalage fournit les points un par un (cf. fig. ci-dessous).

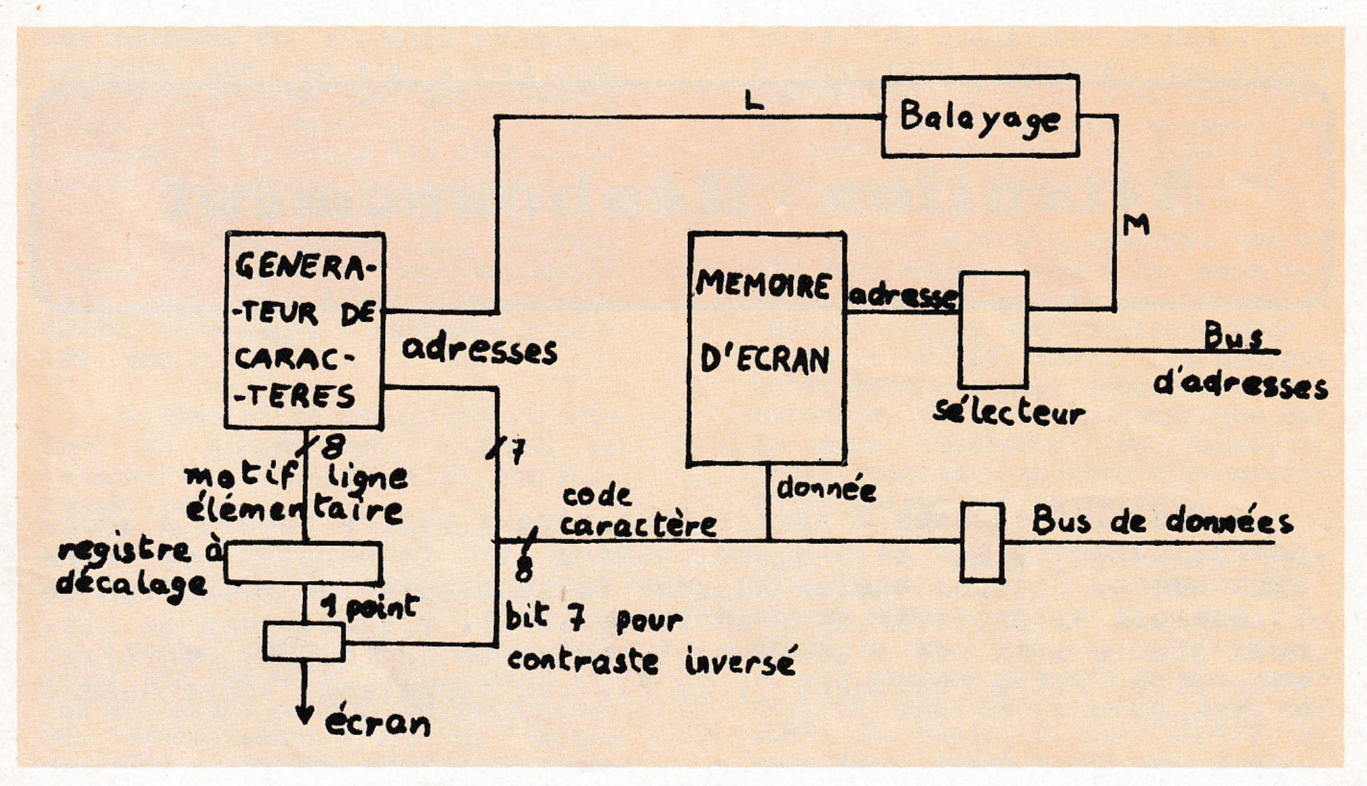


Schéma très simplifié de l'affichage
dans un CBM 3000

Le VIC et les CBM présentent deux différences qui vont jouer fortement pour l'obtention de la haute résolution.

1 - Dans le VIC un caractère et son correspondant en contraste inversé sont présents tous les deux dans le générateur de caractères. Sur PET/CBM, par économie, seul le caractère direct est dans le générateur de caractères : le bit 7 du code caractère va, à part, commander l'inversion de contraste. Seuls les bits 0 - 6 du code caractère vont au générateur.

2 - Dans le VIC, le générateur de caractères est dans l'espace adressable du système : G(=32768 à la mise sous tension) est une adresse du système. Vous pouvez y accéder par PEEK. Mais il y a plus: vous pouvez dire au VIC que G a une autre valeur, et en particulier placer le générateur en RAM. Cela permet la haute résolution comme nous le montrerons dans un prochain numéro.

Au contraire sur CBM, le générateur de caractères n'est pas dans l'espace adressable : vous ne pouvez pas y accéder ni par PEEK, ni en langage machine : vous voyez sur le schéma ci-dessous que ce n'est pas le bus d'adresses du CBM qui fournit les lignes adresses du générateur de caractères.

Pour faire de la haute résolution, il y a alors deux solutions :

1 - réincorporer le générateur de caractères dans l'espace adressable et y substituer de la RAM pour revenir à une solution du type VIC. Le fait que le bit 7 du code soit traité à part crée une petite difficulté, mais cette solution est possible.

2 - revenir en amont du codage par caractères et générateurs de caractères en étendant la mémoire d'écran. C'est cette solution que nous avons adoptée sur un CBM 3000 et qui sera décrite dans un prochain numéro.

A SUIVRE...

Daniel-Jean DAVID

Attention - Réabonnement

Pour les abonnés à partir du numéro 1 (votre numéro d'abonnement commence par 81-1-), ce numéro est le dernier de votre abonnement. Si vous voulez assurer la continuité de réception de la Commode et profiter de l'ancien prix (il y a beaucoup de risques que le prix de l'abonnement augmente au SICOB) renvoyez-nous vite votre règlement (120 F), en rappelant votre numéro d'abonné.

Choisissez votre ROM

Du "HARD" sur CBM petit écran 3000/4000 pour vous simplifier la vie.

Voici une description des deux produits "Hard" que j'ai réalisé pour CBM petit écran de façon professionnelle.

Etant particulier, je ne peux facturer mes interfaces à aucune société.

Ces 2 extensions répondent aux caractéristiques générales suivantes :

* Facilité d'installation : enfichage direct sans soudure sur les supports de circuits intégrés. Certains CBM ont été livrés avec des ROMS soudées aux emplacements $F000 à $9000. Dans ce cas, il faut dessouder à la pompe les ROMS existantes pour mettre des supports.

* La réalisation est "professionnelle" : circuit imprimé double face trous métallisés, connecteurs dorés, inverseurs à contacts argentés, etc.

* Conception modulaire permettant un encombrement minimum et laissant toute possibilité de juxtaposition.

* 1 * CARTE SELECT ROM UNITAIRE *

Cette carte est utilisable aussi sur CBM 8000. L'inverseur situé à l'avant permet de sélectionner manuellement l'une ou l'autre des ROMS ou EPROMS montées verticalement.

Exemple : Sur un CBM 3000, on peut installer sur le support $9000 (UD3) une EPROM du commerce comme VISICALC avec une EPROM personnelle contenant le DOS, HARDCOPIE et EXTRAMON par exemple.

* 2 * INTERFACE BASIC 3 4 AVEC 2 SELECT ROMS *

Cette interface sélectionne automatiquement 5, 6 ou 7 ROMS ou EPROMS situées de $9000 à $F000 UD3 à UD9 , par un simple inverseur en façade. Cela permet de choisir de travailler en BASIC 3 ou 4 avec rappel éventuel par diodes lumineuses LED. A la mise sous tension on commute automatiquement les 5 ROMS du BASIC 4 ou bien les 4 ROMS du BASIC 3 plus la ROM EDEX.

Les 2 derniers supports UD4 et UD3 sont commutables à la main par un inverseur situé sur la carte. (même principe que ci-dessus pour la carte SELECT ROM). Ils peuvent aussi être commutés séparément ou ensemble de façon automatique par le même inverseur en façade. (Positionnement correct d'un micro-interrupteur sur la carte).

NOTE 1 : Cette interface peut être montée pour un CBM 4000 GRAND ECRAN. Alors les ROMS du BASIC 4 sont seules utilisables, car la gestion d'écran de cet appareil ne peut fonctionner avec les ROMS du BASIC 3. Par contre, si vous êtes assez fort pour créer votre propre langage et le mettre

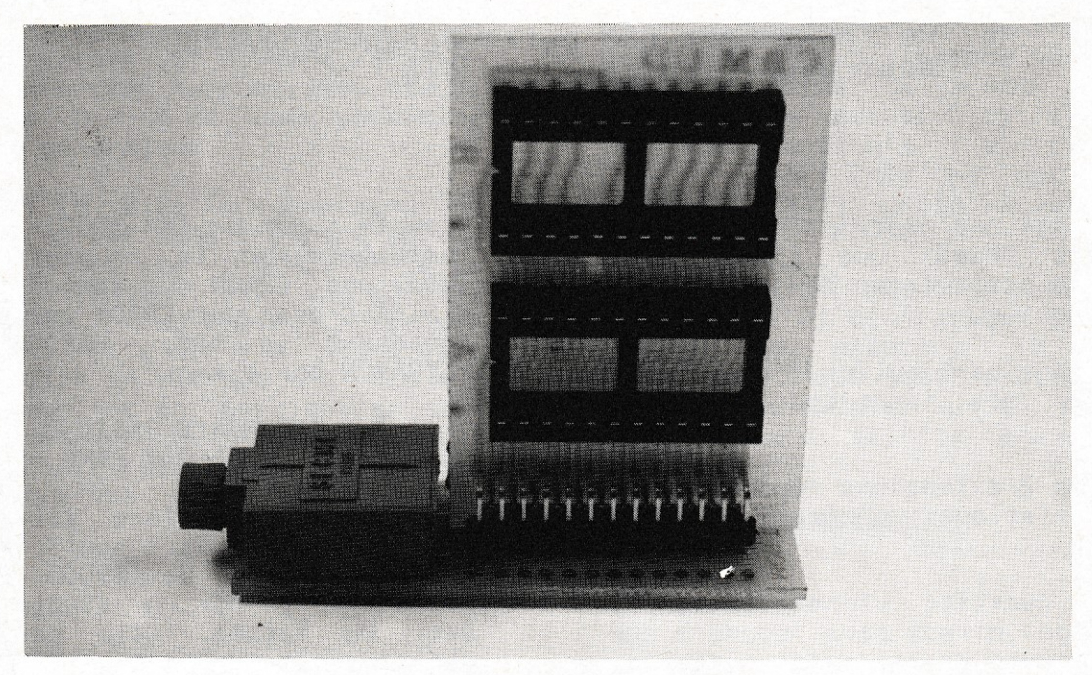
en EPROMS, l'avenir vous reste ouvert.

NOTE 2 : Pour éviter de percer la façade, l'inverseur peut être monté sur un boitier déporté. Mieux encore ! Vous pouvez commuter le BASIC automatiquement par programme en branchant les fils de l'inverseur sur le PORT UTILISATEUR.

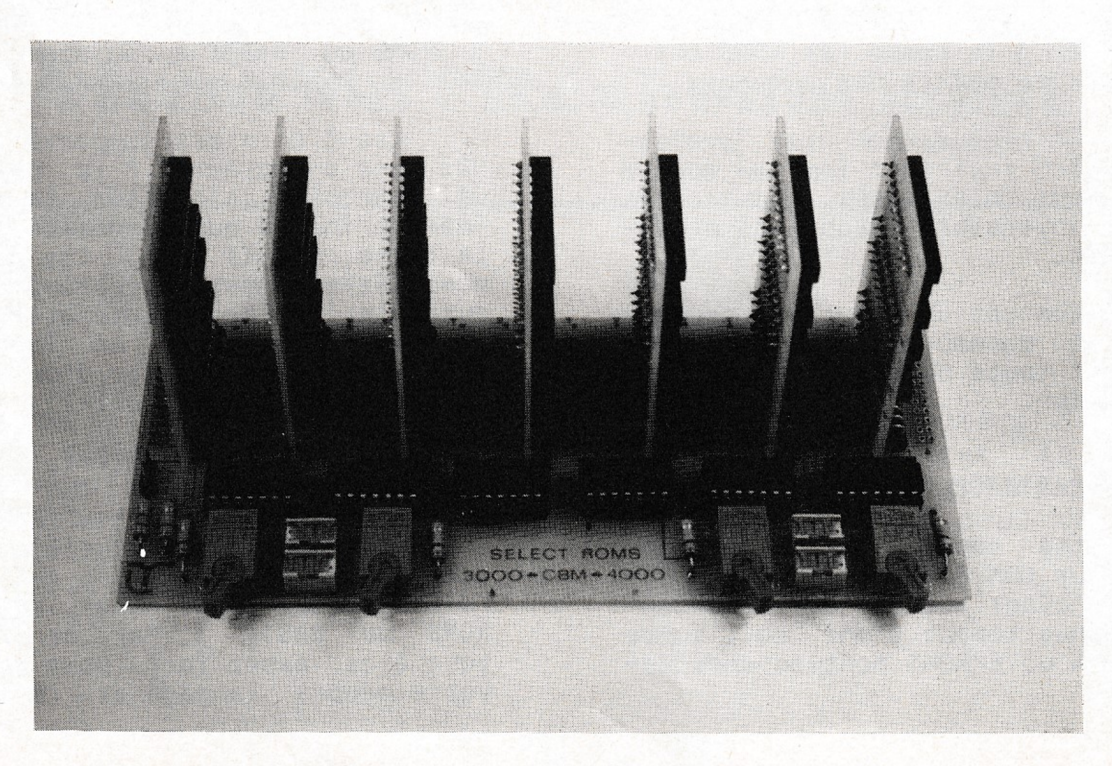
UNE NOTICE DETAILLEE PERMET UNE INSTALLATION EN TOUTE QUIETUDE POUR TOUS LES CAS DE FIGURES.

Pierre THERY

(NDLR : avis à un éventuel distributeur)



Carte Select ROM unitaire



Système BASIC 2 - BASIC 4

Bibliographie

"Mise en oeuvre du BUS IEEE 488"
par G. BASTIDE et J.R. VELLAS.
Editions TESTS

Enfin un livre clair et précis sur le BUS IEEE 488 écrit en Français. Il ne s'agit pas d'une traduction de la norme, mais d'un ouvrage destiné aux utilisateurs. De nombreux exemples de programmes pour calculateurs HP85, HP98XX, PET/CBM, Apple et AIM65 le prouve, car ils traitent, sans exception, toutes les possibilités prévues par la norme. D'autre part, pour ceux qui désirent utiliser la norme IEEE 488-1978, cet ouvrage est une bonne introduction. Mais pour 90 % des utilisateurs sa lecture suffit.

Deux chapîtres sont consacrés à la réalisation d'interfaces par l'utilisateur ; dans ces chapîtres, l'utilisation du circuit MOTOROLA 68488 est donné pour un microprocesseur 6502 et les programmes sont donc utilisables facilement sur les machines COMMODORE !

En conclusion, c'est un livre fort bien fait qui doit devenir l'outil de travail de tout utilisateur du BUS IEEE 488.

Jean-Luc DESCHAMPS

*
* *

LA PRATIQUE DU PET/CBM
D.J. DAVID
Volume I -Périphériques et gestion des Fichiers.

P.58 : CANAL DE COMMANDE :

Après OPEN 8,8,15,il faut faire PRINT #8,"chaîne de Commande" et non PRINT#15,"chaîne de Commande" comme indiqué. (cette coquille est particulièrement déroutante pour le débutant, à cet endroit).

P.60 : VALIDATE :

il n'est pas mentionné que, si cette commande supprime les "fichiers pendants", elle détruit également tous les fichiers "accès direct" du même drive : donc attention !

P.61 : Commande COPY :

contrairement à ce qui est indiqué, on peut utiliser comme fichier destination le nom d'un fichier déjà existant : c'est même de cette façon que l'on peut concaténer 2 fichiers :

```
Exemple : OPEN 8,8,15 :
          PRINT#8,"C1:TATA=1
          :TATA,0:JUJU":CLOSE8.
```

remplace le fichier TATA qui existait sur le drive 1 par un nouveau fichier de même nom, somme de

l'ex-TATA et de JUJU du drive 0.

P.67 : FIN de fichier :

Contrairement à ce qui est indiqué, la variable d'état ST peut parfaitement être utilisée pour tester la fin du fichier, où elle prend la valeur 64 (au moins dans le cas de mon ensemble 3032 + 8050!) : point n'est besoin d'utiliser un indicateur du type $$$.

Exemple :

```
100  OPEN 85,8,5
110  INPUT #85,A$
120  IF ST <>64 THEN PRINT A$:
     GOTO 110
130  PRINT "TERMINE!":CLOSE 85
```

Ce n'est peut-être pas orthodoxe, mais c'est bien pratique !

J. PIERRAT



LE FEED-BACK DE LA COMMODE

Pour produire une revue qui vous donne toujours plus de satisfaction, nous avons absolument besoin de vos réponses à ces questionnaires.

Il y a un questionnaire pour chaque numéro. Parmi les réponses concernant chaque numéro, deux seront tirées au sort et leurs auteurs recevront un petit souvenir de La Commode.

Comme au patinage artistique, vous attribuez deux notes à chaque article: une note sur l'intérêt du sujet et une note sur la façon dont le sujet est traité (rédaction, clarté, etc.).

Voici les barêmes :

sujet :		traitement :	
intérêt capital................	4	exellent................	4
très intéressant...............	3	bon.....................	3
assez intéressant..............	2	moyen...................	2
pas d'intérêt pour moi.........	1	mauvais.................	1
n'a pas, d'après moi , sa place dans La Commode................	0	nul.....................	0

Nous donnons ici quatre questionnaires, depuis le numéro 1. Il y aura ultérieurement, un seul questionnaire portant sur le numéro considéré.

Numéro 1 : titre	note sujet	note traitement
La gamme Commodore		
Visite au PET Show		
Premier contact avec le VIC		
Le CBM : un ordinateur polyglotte		
COMAL : le dernier-né des langages		
Le PASCAL du CBM		
Comment accélérer l'exécution de vos programmes BASIC		
Mini annuaire téléphonique		
PASCAL, un utilitaire de référence croisée		
Spooling disque-imprimante		
L'interface sonore sur le PET/CBM		
Adresses absolument fondamentales		

titre	note sujet	note traitement
Le jeu du pendu		
Adresses des routines IEEE		

Autres commentaires :...
....
....

Numéro 2 titre	note sujet	note traitement
Courrier des lecteurs		
Visite au Sicob		
Interview Europrocessing		
Les multiples facettes de l'instruction PRINT		
Calcul des primes de remboursement d'un prêt		
Un utilitaire : LIBRARIAN		
Programme de référence croisée en BASIC		
Banc d'essai du lecteur de cartes Chatsworth		
Cahier spécial comptabilité : introduction.		
Les différents modèles d'unités centrales		
Les unités de disques		
Les imprimantes		
Le modèle 43032		
Comment augmenter la mémoire de votre 3008 ou 3016		
Comment adapter un programme de 3000 en 4000		
Comment transférer un fichier d'une unité de disques sur une autre		
Correspondance des points d'entrée BASIC 1,2 et 4		
Adresses du CBM série 4000/8000		
Le bus IEEE 488		
Jeu de tennis		
Jeu du tir à la cible		
Ephémérides astronomiques		
Registres des boîtiers d'entrée/sortie		
Notre cadeau de Noàel		

Autres commentaires :...
....
....

Numéro 3 titre	note sujet	note traitement
Actualité		
Courrier des lecteurs		
Bourse d'échange des programmes		
Les monodisques 2031 et 1540		
Un programme de tri		
Bricolage : un bouton de RESET		
Le programme PLATO'S		
Le coin du débutant		
Le point sur le VIC		
PMLM-VIC		
LIRARIAN-VIC		
Changements d'adresses dans le VIC		
Les interfaces du VIC		
MASTERMIND sur le VIC		

titre ----- | note sujet | note traitement

Routines-système du VIC
Manipulations magnétos
Banc d'essai du jeu d'entreprise INFODEV
Utiliser OZZ
Jeu Piranha

Autres commentaires :...
....
....

Numéro 4 titre | note sujet | note traitement

Actualité
Courrier des lecteurs
Premiéres impressions sur le CBM9000
Mieux utiliser BASIC
Petits trucs et combines
Random en BASIC 4.0
Opérations sur les fichiers
Le jeu de JOSEPHUS
Le tampon clavier
Registres d'entrées/sorties du VIC
Connectez deux magnétos à votre VIC
Adresses stratégiques du VIC
VIC A BRAC
Affichage haute-résolution
Choisissez votre ROM

Autres commentaires :...
....
....

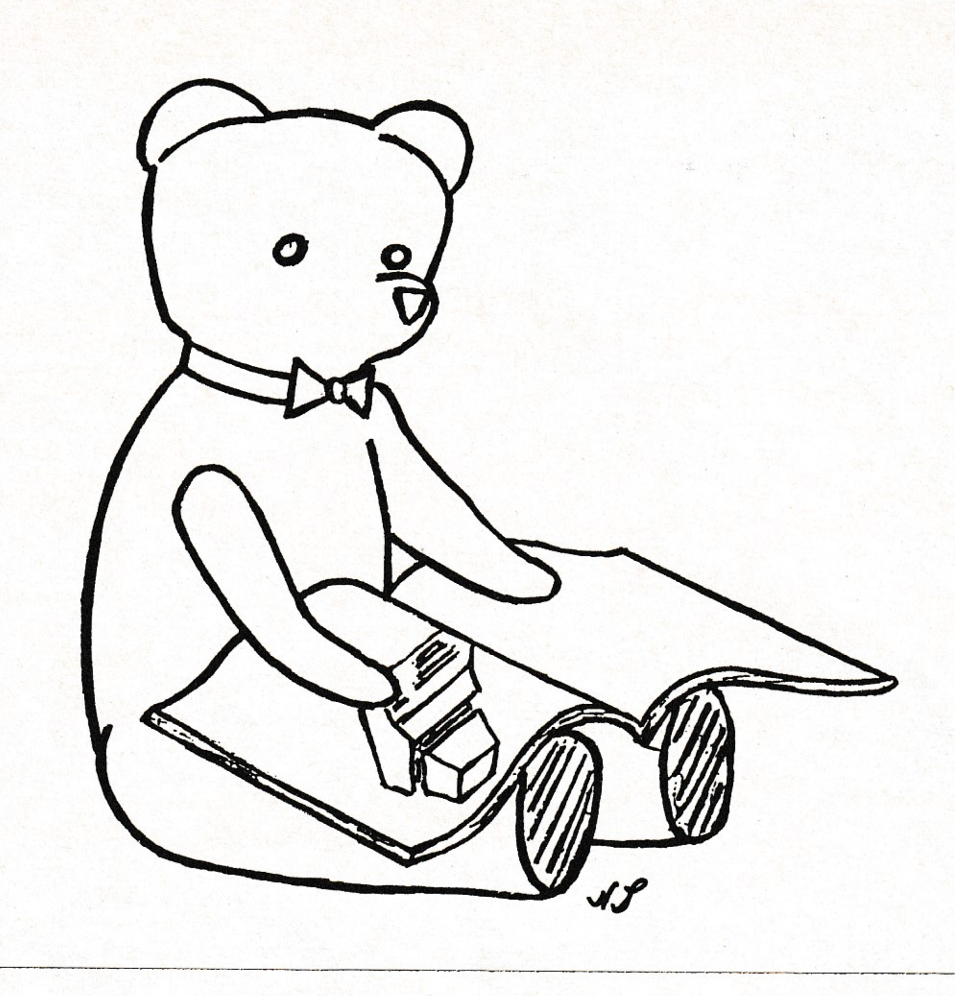
Petit questionnaire
(réponses indispensables pour être tiré au sort).

nom :..............................
adresse : ....
....
âge :.......
Profession :.......................

J'utilise un Commodore modèle...
pour :
ma profession mes loisirs

Précisez SVP, si vous utilisez différents modèles.

A renvoyer à La Commode (service FB)
28, rue Vicq-d'Azir
75010 PARIS





édi tests LA
BIBLIO·TECHNIQUE
DES PROFESSIONNELS
les SYSTEMES à microprocesseurs
mise en œuvre du BUS IEEE 488
édi tests
MISE EN ŒUVRE DU BUS IEEE 488. Utilisation et réalisation d'appareils. Par Gérard Bastide et Jean-René Vellas.
Plus de mille appareils sont équipés en IEEE 488. Ce livre décrit comment mettre en œuvre toutes les possibilités du BUS IEEE, il comprend la description et les syntaxes sur des calculateurs différents de toutes les commandes unilignes ou multilignes, universelles ou adressées et la réponse à toutes sortes de questions : comment connaître au premier coup d'œil les capacités d'un périphérique ? Deux appareils peuvent-ils communiquer sans requérir l'intervention ou même la présente du calculateur ?...
LES SYSTÈMES A MICROPROCESSEURS. Par Daniel-Jean David.
Ce livre est une initiation aux conditions techniques de la révolution micro-informatique. Les différents circuits intégrés : microprocesseurs, mémoires, boîtiers d'entrées-sorties sont décrits ainsi que la façon de les assembler pour former un système. Les phases du traitement d'une application et du développement d'un système à microprocesseur sont décrites, notamment du point de vue du logiciel (programmation en assembleur) et des choix à effectuer.
Chaque volume : 128 pages 75,00 FF / 570 FB
AGRAPH

Petites Annonces - Clubs

CBM 3032, cherche contact sur région Ugine, Albertville, Faverges, en vue d'échanges de programmes, de trucs, d'idées et peut être pour créer un club Commodore 73 (Savoie).

Je recherche également des gens utilisant le langage Forth 2.0, pour avoir des tuyaux sur son utilisation.

BRUGEASSOU Jacques, Villa 110
Les Fontaines 73400 UGINE

*
* *

Les membres du club Magenta souhaitent que vous insistiez sur l'importance de l'existence de la touche "shift-lock" sur les micros-ordinateurs.

En effet, certaines personnes utilisent ces appareils d'une seule main ou même d'un seul doigt.

Merci pour eux.

Le club Magenta organise des cours de Basic gratuits au mois de juillet.
Adresse : 18, boulevard de Magenta
75010 PARIS
Tél. : 208.69.29

Le club Magenta cherche à avoir des contacts avec les autres clubs et les journalistes des revues d'informatique.

Rappelons que le club fonctionne tous les samedis de 18 h à 20 heures.

*
* *

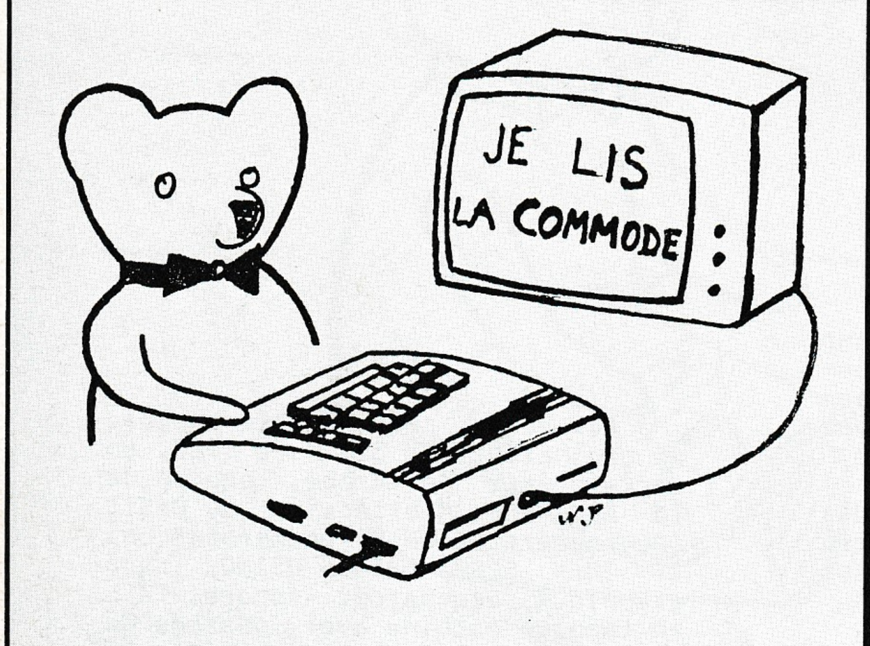
Je vends, cause double emploi, un ordinateur CBM 32K Mars 1980, en parfait état de marche, équipé de la ROM Edex d'extension du BASIC (Autonumérotation, Renumérotation, IF THEN ELSE, PRINT USING, Tracé graphique, Générateur sonore, CALL en langage machine avec passage de paramètres...), de son magnéto K7 et de sa documentation d'origine pour 7000 Frs. à débattre.

J'offre de plus THE PET REVEALED, ouvrage de référence, qui dévoile tous les secrets du CBM : très utile pour celui qui veut réellement comprendre comment fonctionne un micro-ordinateur. On y trouve entre autre :
- Codage des programmes BASIC en mémoire centrale.
- Adresses des routines arithmétiques en flottant (+,-,*,/,LOG,SIN, INT...) qui simplifient le travail de celui qui programme en langage machine.
- l'Assembleur du 6502.

J'offre un LOGICIEL UTILITAIRE (ASSEMBLEUR et DESASSEMBLEUR 6502) ainsi qu'un LOGICIEL de JEUX (Echecs, Bridge, Super-Invaders, Star-Force,Star-Trek...)

Si cette proposition vous interesse, joindre rapidement :

Gérard AVICE
1, rue de Colombes
92400 COURBEVOIE
Tél. : 16-(1)-788-88-73



Rédaction

Rédacteur en chef :
Daniel TRECOURT

Chefs de rubriques :
Jacques COUTHURES
Jean DELAVILLE
Jean-Luc DESCHAMPS
Pierre-Etienne THALBERG

Secrétaire de rédaction :
Chantal Deschamps

Directeur de la publication :
Daniel-Jean DAVID

Rédaction-vente-abonnements :
28, rue Vicq d'Azir
75010 PARIS

Diffusion :
Ed. du P.S.I.
41-51, rue Jacquard
BP 86
77400 LAGNY

Publicité :
FORCE 7
41, rue de la Grange aux Belles
75483 PARIS CEDEX 10
Tél. 238.66.10

Publié par SEDERMI SARL
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Bulletin d'abonnement

A envoyer à La Commode
28, rue Vicq d'Azir - 75010 PARIS

Je désire m'abonner à la Commode (1 an, 4 numéros) à partir du numéro ....

NOM : .... PRÉNOM : ....

Adresse : ....
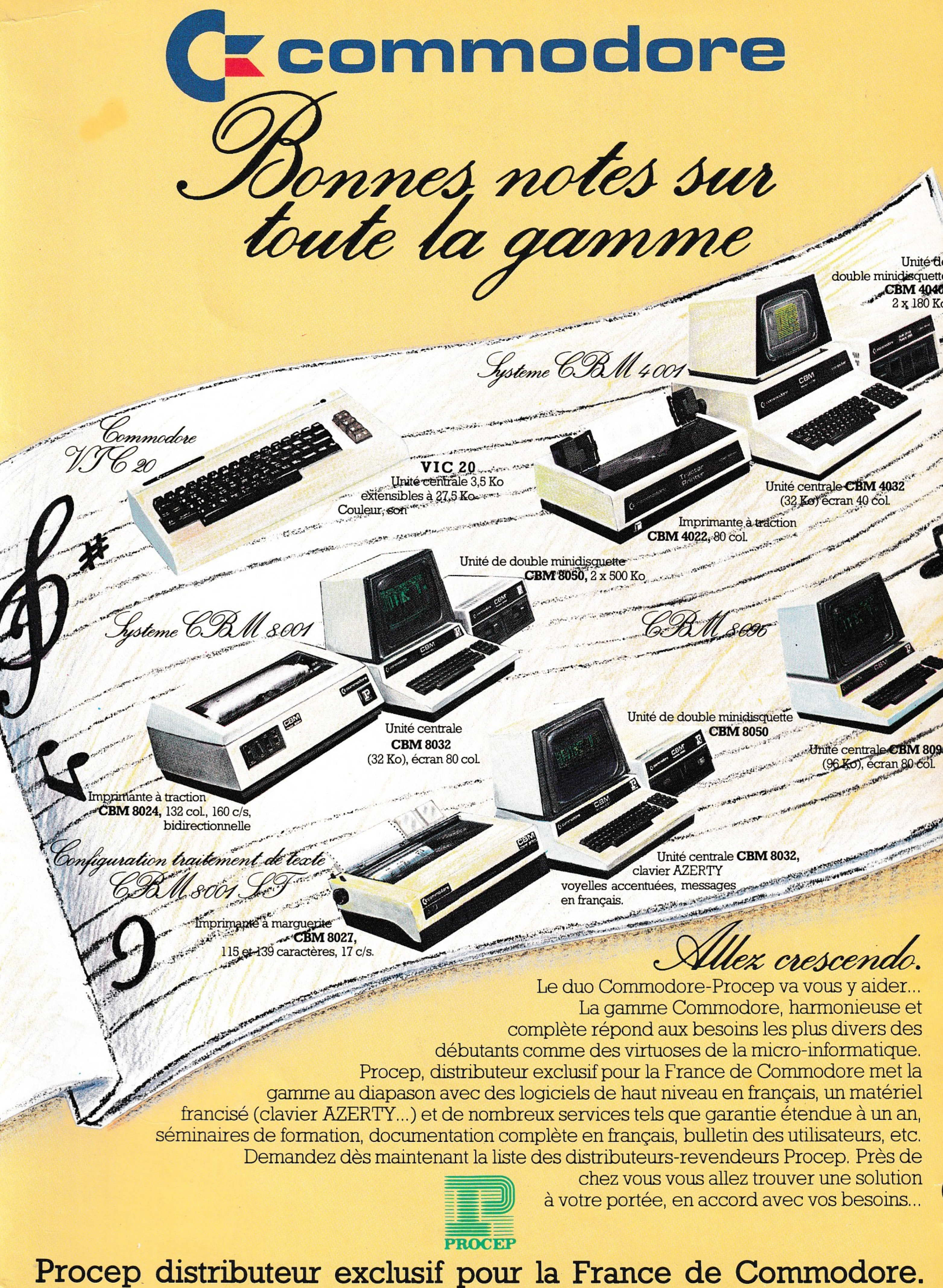
....

Signature

Règlement 120 F à l'ordre de SEDERMI :

☐ CCP ☐ CB ☐ ESPECES

Imprimerie Laballery, 58500 Clamecy. Imprimé en France. Directeur de la publication Daniel-Jean David
Dépôt légal : juillet 1982. Numéro de Commission paritaire : en cours. ISSN 0292-5079


commodore
Bonnes notes sur toute la gamme
Unité de double minidisquette CBM 4040 2 x 180 Ko
Système CBM 4001
Commodore VIC 20
VIC 20
Unité centrale 3,5 Ko extensibles à 27,5 Ko. Couleur, son
Unité centrale CBM 4032 (32 Ko) écran 40 col.
Imprimante à traction CBM 4022, 80 col.
Unité de double minidisquette CBM 8050, 2 x 500 Ko
Système CBM 8001
CBM 8096
Unité centrale CBM 8032 (32 Ko), écran 80 col.
Unité de double minidisquette CBM 8050
Unité centrale CBM 8096 (96 Ko), écran 80 col.
Imprimante à traction CBM 8024, 132 col., 160 c/s, bidirectionnelle
Configuration traitement de texte CBM 8001 SF
Unité centrale CBM 8032, clavier AZERTY voyelles accentuées, messages en français.
Imprimante à marguerite CBM 8027, 115 et 139 caractères, 17 c/s.
Allez crescendo.
Le duo Commodore-Procep va vous y aider... La gamme Commodore, harmonieuse et complète répond aux besoins les plus divers des débutants comme des virtuoses de la micro-informatique. Procep, distributeur exclusif pour la France de Commodore met la gamme au diapason avec des logiciels de haut niveau en français, un matériel francisé (clavier AZERTY...) et de nombreux services tels que garantie étendue à un an, séminaires de formation, documentation complète en français, bulletin des utilisateurs, etc. Demandez dès maintenant la liste des distributeurs-revendeurs Procep. Près de chez vous vous allez trouver une solution à votre portée, en accord avec vos besoins...
PROCEP
Procep distributeur exclusif pour la France de Commodore.
19-21, rue Mathurin-Régnier - 75015 Paris - Tél. : 306.82.02 - Télex : 204 875 F.

Pour mieux choisir "votre" ordinateur et pour mieux l'utiliser.



Lisez

L'ORDINATEUR INDIVIDUEL

Vous y trouverez :

L'actualité et les tendances de l'informatique individuelle • des galops et des bancs d'essai des principaux matériels • des panoramas et des tests comparatifs • le point des grandes manifestations internationales • des articles d'initiation • des synthèses • des programmes • des interviews "exemplaires" • des conseils • des idées • des astuces.



chez votre marchand de journaux

41 rue de la Grange aux Belles - 75010 Paris